Forecasting and Optimization in Modern Cosmology

Jacques Kotze

Thesis presented for the degree of

DOCTOR OF PHILOSOPHY

in the Department of Mathematics and Applied Mathematics

at the

University of Cape Town

June 2010

Declaration

The work presented in this thesis is partly based on collaboration with Bruce Bassett, Renée Hložek, Yabebal Fantaye, Mike Brownstone, Antonio Cardoso, Marina Cortês and Patrick Okouma. The list below identifies sections which are partially based on the listed publications (or article in preparation):

- Chapter 4 includes figures and text from [1].
- Chapter 1 & 5 along with Appendix B includes figures and text from [2]
- Chapter 6 and Appendix C includes figures and text from [3].

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for degree and that it represents my own work.

Signed by candidate

Jacques Kotze

Forecasting and Optimization in Modern Cosmology

Jacques Kotze

Thesis presented for the degree of

DOCTOR OF PHILOSOPHY

in the Department of Mathematics and Applied Mathematics

at the

University of Cape Town

June 2010

Declaration

The work presented in this thesis is partly based on collaboration with Bruce Bassett, Renée Hlozek, Yabebal Fantaye, Mike Brownstone, Antonio Cardoso, Marina Cortês and Patrice Okouma. The list below identifies sections which are partially based on the listed publications (or article in preparation):

- Chapter 4 includes figures and text from [1].
- Chapter 1 & 5 along with Appendix B includes figures and text from [2]
- Chapter 6 and Appendix C includes figures and text from [3].

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for degree and that it represents my own work.

Jacques Kotze

Abstract

Cosmology is emerging into a new and exciting period thanks to a wealth of ongoing and planned massive surveys which will deliver exponentially growing volumes of data over the next two decades. As a result of this rapid growth, which exhibits fierce competition between different surveys due to the spiralling costs, forecasting and optimization have become critical to help best use and benefit from this new boon.

In this thesis various aspects of forecasting and optimization are explored, with particular emphasis on, but not limited to, cosmology. We introduce a new optimization algorithm which significantly outperforms all standard algorithms, especially in higher dimensions where the improvement is remarkable. The new algorithm, Hybrid-MTM, should provide a powerful new tool in addressing high-dimensional optimization problems.

We then forecast the prospects for detecting dynamics in tracking dark energy models. We show that Big Bang Nucleosynthesis and Cosmic Microwave Background constraints in these models are extremely difficult to match with existing data. As a result it is unlikely that a detectable deviations from the cosmological constant for these models is possible before the Stage-IV DETF experiments, which will only come on-line post-2015.

Finally we present new results on Fisher matrix forecasts for cosmology produced using the Fisher4Cast code. Fisher4Cast allows novel insights into the nature of how information is gained from cosmological experiments and the interplay between the measurements of Hubble, distance and growth in constraining cosmological parameters. In the final chapter we provide a detailed overview of the code structure in Fisher4Cast and its Graphical User Interface together with its unique features including a LaTeXreporting module which breaks new ground in the automated generation of publication-quality scientific research.

Acknowledgements

I would like to thank my supervisor, Prof Bruce Bassett, for teaching me a great deal and giving me valuable insight along with forever shaping my perspective on research, the world and life. I would like to thank the gifted students of Team Fisher for a great work experience and learning exactly what the virtues of team work can deliver.

A special thanks to the staff and students of the ICG. To Bob Nichol and Roy Maartens for hosting me and taking such a keen interest in my welfare and progress. To Dan Carson, Ben Hoyle and Cristiano Sabui for their help in completing this thesis and for the K's we shared.

To my parents for their unwavering love and support. To Lauren, for enduring the tough times with me and bring the measure of gentleness, beauty and humour into my life that I needed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Statistics and Optimization

"Statistics means never having to say you're certain."
Unknown

The first half of this chapter introduces basic statistical elements leading up to Fisher Matrix formalism, which forms a focal point for future chapters present in this thesis. The second half of this introduction addresses the concept of optimization and presents some standard heuristic algorithms which are used and expanded on in Chapter 3.

1.1 Statistics

1.1.1 Random Variables

Random variables are usually denoted by X and are used in probability theory to assign a numerical value to the outcome of a random process as depicted in Figure 1.1. An example of this would be flipping an unbiased coin, the two possible outcomes of this random process are either heads or tails. In this instance you can assign a one if the coin lands on heads or a zero if it lands on tails:

$$X = \begin{cases} 1 & \text{Heads}, \\ 0 & \text{Tails}. \end{cases} \tag{1.1}$$

There are two types of random variables, discrete and continuous. The previous case of flipping a coin is an example of a discrete random variable where you can have only one of two possible outcomes. A continuous variable would be used to quantify the

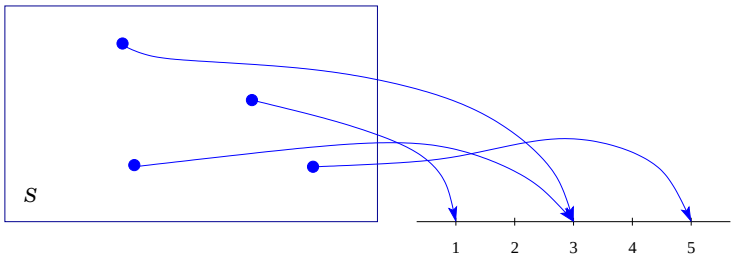


Figure 1.1: **Discrete Random Variable** – Here we show how a discrete random variable can be seen as a process of assigning a finite set of random events from a space of all possible outcomes, S, sometimes called the experiment space, to a discrete set of numerical values. We typically deal with the continuous case.

amount that it will rain on a given day. There are, in principle, an infinite number of measurements that could be observed when taking the final measurement of rain.

1.1.2 Probability Density Function

The relative likelihood of a random variable occurring at a given point in the sample space, can be quantified by a function known as the probability density function[1] (PDF), $f_X(x)$.

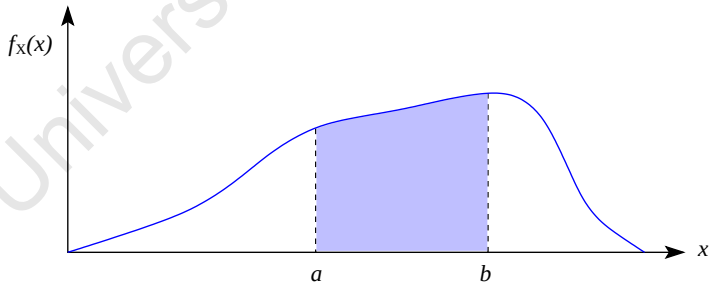


Figure 1.2: **Probability Density Function** – The PDF, $f_X(x)$, for a random variable, x, is shown with the shaded area between $[a, b]$ representing the probability, $P(a \leq x \leq b)$, as given in Eq. (1.3).

The principle law of probability ensures that the integral over the entire range of outcomes must equal one [17–19], i.e. one event within the space of all possible

[1] It is called a probability distribution function in the discrete case.

outcomes is guaranteed to occur:

$$\int_{-\infty}^{\infty} f_X(x)dx = 1. \tag{1.2}$$

It follows that the probability of an event occurring within a specified range is then given by,

$$P(a \leq x \leq b) = \int_a^b f_X(x)dx, \tag{1.3}$$

which is illustrated in Figure 1.2.

1.1.3 Expectation Values

An intuitive understanding of the expectation value can be gained by using the *law of large numbers* which loosely states that the expectation value converges to that of the sample average as the total number of samples, n, tends to infinity, we obtain:

$$\mu \equiv \langle X \rangle \approx \sum_{i=1}^{n} \frac{x_i}{n}, \tag{1.4}$$

where μ indicates the mean and the angle brackets the expected value which is also often written as $E(X)$.

Another useful interpretation of the expectation value can be formulated as a weighted average by the probability of each given random variable. In the discrete case this would be a sum of the random variables weighted by the probability of each random variable occurring. In the continuous case it is the PDF, $f_X(x)$, weighted integral of the random variables:

$$\langle X \rangle = \int_{-\infty}^{\infty} x \ f_X(x)dx, \tag{1.5}$$

as given in [18], or for an expectation value of a function $g(X)$,

$$\langle g(X) \rangle \equiv \int_{-\infty}^{\infty} f_X(x)g(x)dx. \tag{1.6}$$

The result for an expectation value represents a long term average and may thus be unphysical, such as an expected value of 2.5 children per household in a census survey.

The elements of a Fisher Matrix, discussed in Section 1.1.8, are each expectation values and thus underline their importance when attempting to understand the Fisher Matrix framework.

1.1.4 Variance & Covariance

A measure of the statistical dispersion of a single random variable, X, around its expectation value (mean value), $\mu = \langle X \rangle$, is called the variance,

$$\mathrm{Var}(X) = \sigma^2 = \langle (X - \mu)^2 \rangle. \tag{1.7}$$

The positive square root of the variance is known as the standard deviation, σ. This allows us to quantify how "scattered" the data is around the mean of a random variable.

Figure 1.1.4 shows the standard deviation, σ, for a specific example where the distribution of the error or noise around the mean value is a Gaussian. This illustrates the associated confidence interval each σ level produces.

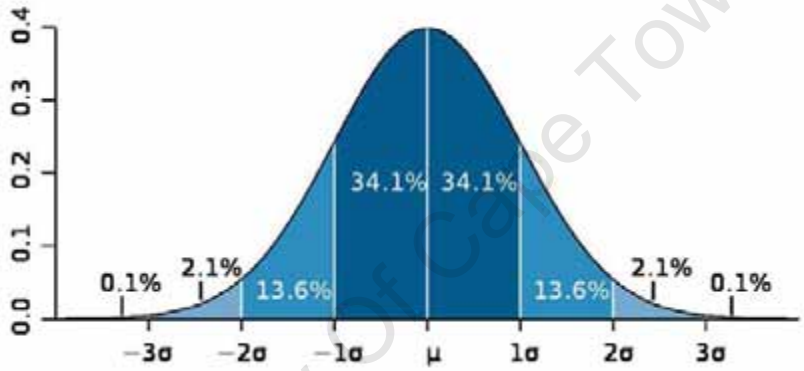


Figure 1.3: **A Gaussian Distributed Error** – Here a Gaussian distribution is shown around the mean value, μ. Each σ is shown with its associated confidence level. The 1σ confidence level corresponds to $\sim 68\%$ while 2σ represents a 95% confidence level (Credit[2]).

The covariance is a generalisation of the variance where the variation of a set of n random variables, $\{X_1, X_2, \ldots, X_n\}$, around their respective mean values, $\{\mu_1, \mu_2, \ldots, \mu_n\}$, are taken as opposed to a single variable around its mean. The covariance for n number of random variables is thus given by a matrix,

$$C_{ij} = \langle (X_i - \mu_i)(X_j - \mu_j) \rangle, \tag{1.8}$$

where $i = 1, 2, \ldots, n$ and the variance is a special case where both variables are identical ($i = j$). Σ is also often used as an additional symbols to refer to the covariance matrix above.

[2] commons.wikimedia.org/wiki/File:Standard_deviation_diagram.png .

If the covariance between two random variables is zero, then they are uncorrelated or independent. A diagonal covariance matrix thus implies that the variables are uncorrelated and only the variance, σ^2, remains. The data covariance matrix is important in defining a general chi-squared statistic for a correlated Gaussian likelihood, done in Section 1.1.6, which is a key step that is required when defining the Fisher Matrix, Section 1.1.8.

1.1.5 Probability and Bayes' Theorem

In this subsection we introduce a basic guide to probability and build towards a brief overview of Bayes' theorem. Venn diagrams are used as an intuitive crutch to grasp the idea of probability. In a Venn diagrams the area of an event, e.g. X, is representative of the probability of that event occurring, $P(X) \equiv \text{area}(X)/\text{area}(S)$, where S is the space of all possible outcomes (also called the experimental space). Figure 1.4 lists some examples of Venn diagrams and their equivalent probabilities.

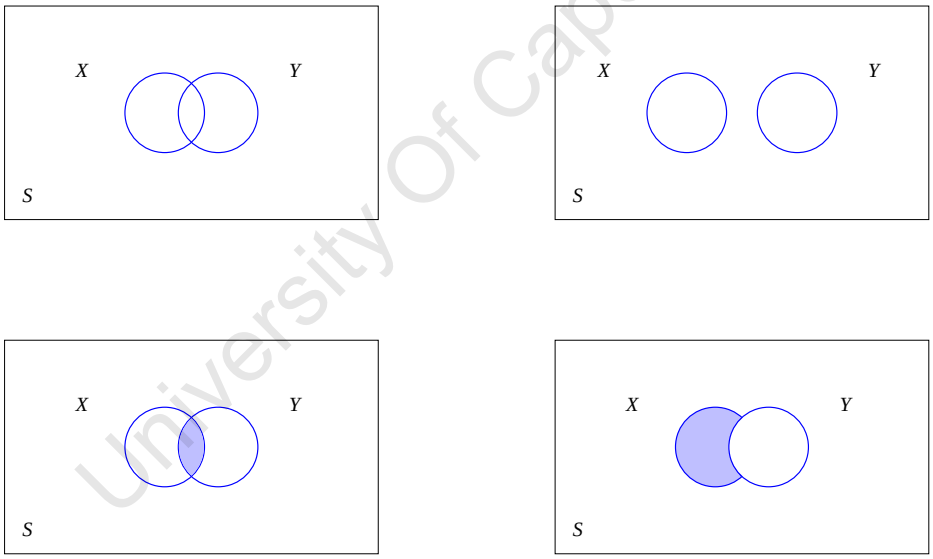


Figure 1.4: **Venn Diagrams** – This set of Venn diagrams illustrates a few simple examples that are representative of a probability or event in a space of all possible outcomes, S. Random variables X and Y in space S (top left). Mutually exclusive events X and Y (top right). The shaded region is equivalent to the joint probability $P(X \cap Y)$ (bottom left). The shaded region is the same as $P(X \cap Y')$ where Y' is the complement of Y (bottom right).

- **Joint probability** $P(X, Y)$ or $P(X \cap Y)$ is the probability that two events

happen in conjunction. This is analogous with the Venn diagram $X \cap Y$ in Figure 1.4.

- **Conditional probability** $P(X|Y)$ defines the probability that X will occur given that the event Y has happened. It is read as the probability of X given Y.

- **Marginal probability** is the probability $P(X)$ occurs regardless of the event Y. This can be explicitly shown for two possible events X and Y to be $P(X) = P(X \cap Y) + P(X \cap Y')$, this is known as marginalisation and will be discussed in more detail in Section 1.1.7. Note that Y' is the same as the area excluding the event Y in the Venn diagram, i.e. the area mutually exclusive to the event Y (outside) is given by Y'. This implies that Y' is the complement of Y.

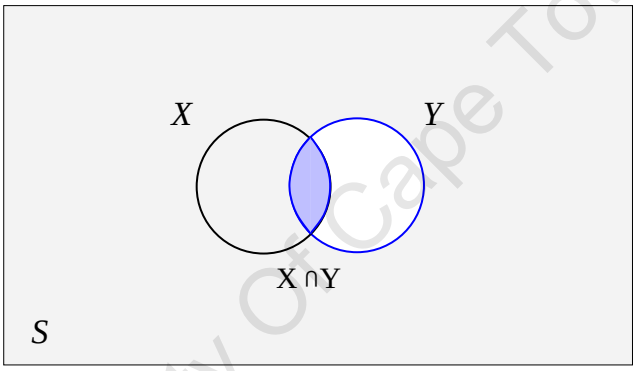


Figure 1.5: **Conditional Probability** – This is the Venn diagram for a conditional probability $P(X|Y)$ where X is conditioned on Y. The usual space of all outcomes, S, is reduced to only be the enclosed subspace of the blue bounded region, Y, which is being conditioned on. Hence one is only interested in that portion of X which is in Y and its probability relative to Y. This is shown by blue shaded region which is representative of $P(X|Y)$ while the grey shaded region shows the region of exclusion which is not taken into account. It should be noted that this Venn diagram of the conditional probability is not equivalent to that of $P(X \cap Y)$ shown in Figure 1.4, since in that case the entire space of outcomes, S, is being considered and here only the conditional subspace (blue bounded area), Y, is considered.

Figure 1.5 shows a specific example of a Venn diagram that is representative of the conditional probability $P(X|Y)$ where X is conditioned on Y. In this case the space of outcomes, usually denoted by S, has been limited to only be Y, producing a result where $P(X|Y) \propto \text{area}(X \cap Y)/\text{area}(Y)$, shown in the Venn diagram of Figure 1.5. Thus the associated probability can be concluded to be $P(X|Y) = P(X \cap Y)/P(Y)$ which can be reduced to $P(X \cap Y) = P(X|Y)P(Y)$. This relationship also follows

directly from the definition of the conditional probability and the multiplication law of probability [17–19]:

$$P(X,Y) = P(X|Y)P(Y). \tag{1.9}$$

Since the reciprocal is also true we can rearrange the above probability,

$$P(Y,X) = P(Y|X)P(X). \tag{1.10}$$

From this simple deduction we immediately can infer Bayes' theorem where $P(X,Y) = P(Y,X)$:

$$P(X|Y) = \frac{P(Y|X)}{P(Y)}P(X), \tag{1.11}$$

as given in [17–19].

In most instances cosmologists would use Bayes' theorem for a mix of data and theoretical parameters. For this reason we use the notation where $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ represents data from an experiment and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$ denotes the parameters of the model under consideration.

$$\underbrace{P(\boldsymbol{\theta}|\mathbf{d})}_{\text{Posterior Probability}} = \frac{\overbrace{P(\mathbf{d}|\boldsymbol{\theta})}^{\text{Likelihood}}}{\underbrace{P(\mathbf{d})}_{\text{Evidence}}} \underbrace{P(\boldsymbol{\theta})}_{\text{Prior}} \tag{1.12}$$

- *Posterior probability* is referred to as the revised probability that is updated and used once new information becomes available. The important point here is that we want $P(\boldsymbol{\theta}|\mathbf{d})$ but are only able to calculate $P(\mathbf{d}|\boldsymbol{\theta})$.

- *Likelihood* is often represented as $\mathscr{L}(\boldsymbol{\theta}) = P(\mathbf{d}|\boldsymbol{\theta})$ which allows us to infer the probability of a parameter given the data.

- *Prior* is a measure of any previous knowledge or belief about the parameters. This would include any known experimental results. In the absence of prior information often a 'flat prior' is used, independent of $\boldsymbol{\theta}$. However if one does parameter estimation with a new set of parameters, the prior for the new parameter, θ, will not be flat in general.

- *Evidence* acts as a normalisation constant for the probabilities. It can be ignored when trying to estimate parameters since it has no dependence on $\boldsymbol{\theta}$.

1.1.6 Parameter Estimation

The situation often arises where experimental data is collected and then needs to be tested with a corresponding model. Since we don't always know how correct a model is we often require the reciprocal where we want to know which parameter values best match the measured data. This was defined in the previous section as the likelihood. The maximum likelihood and the Pearsons chi-square test are common techniques often used for this purpose that we next introduce.

Chi-squared Statistic

The chi-squared statistic determines the extent to which an observation matches the theoretical expectation, for a chosen model. Given the observed data, d_i, with an assumed Gaussian distribution centered around a model expectation, X_i, for a particular parameter, θ_α, the chi-squared is defined as,

$$\chi^2(\theta_\alpha) \equiv \sum_i \Delta_i{}^T \mathbf{C}^{-1} \Delta_i, \tag{1.13}$$

where $\Delta_i \equiv d_i - X_i(\theta_\alpha)$ is a vector and $\mathbf{C}$ is the data covariance matrix. This accounts for the most general case where data are correlated and is revisited when we introduce the Fisher Matrix in Section 1.1.8. If we can assume the data are uncorrelated, then only the diagonal component of the covariance matrix remain. This reduces the above expression to the more familiar form of the chi-squared statistic,

$$\chi^2(\theta_\alpha) \equiv \sum_i \frac{\Delta^2}{\sigma_i^2}, \tag{1.14}$$

where σ_i^2 is the variance for the observed data at index i.

To find the parameter values that best correspond to the observed data we can vary the parameters, θ_α, until the χ^2 is minimized. By minimizing the χ^2 one is also simultaneously maximizing the likelihood, since $\mathscr{L} \propto \exp[-\chi^2/2]$. The idea of maximizing the likelihood is also used in parameter estimation and is discussed next.

Maximum Likelihood

This is a widely used statistical estimator. Within the likelihood function is a mix of parameters and data. The aim in this technique is to maximise the probability of the data, d_i, fitting a parameter, θ_i. The parameters are varied until the maximum likelihood is achieved. These values for the parameters, $\hat{\theta}_i$, are then called the Maximum Likelihood Estimators (MLE). This can be seen as an optimization problem,

introduced and discussed further in Section 1.2, where you are attempting to achieve the maximum likelihood,

$$\mathscr{L}(d_1,\ldots,d_N|\hat{\theta_1},\ldots,\hat{\theta_N}) \geq \mathscr{L}(d_1,\ldots,d_N|\theta_1,\ldots,\theta_N), \tag{1.15}$$

and thus the parameter values that agree most closely to the data provided. One of the benefits of using the maximum likelihood prescript is that the MLE's become the unbiased minimum variance estimators as the sample size increases [17]. The minimum variance estimator produces the lowest variance for any possible values of the parameters considered. On the downside a small sample size can lead to a strongly biased result.

1.1.7 Marginalisation

In Section 1.1.5 the marginal probability is described as the probability that $P(X)$ occurs regardless of the event Y i.e. $P(X) = P(X \cap Y) + P(X \cap Y')$. In this case there is only a single event that is not of interest, Y, called a nuisance parameter.

In a more realistic and general case there are likely to be p parameters of interest and an additional n nuisance parameters that are not of no direct interest but still form a natural part of the problem. As a result we are usually only interested in the probability of the p key parameters, which is obtained by marginalising over the nuisance parameters, *viz.*

$$P(\theta_1,...\theta_p) = \int_{-\infty}^{\infty} P(\theta_1,..,\theta_p,..,\theta_{p+n}) d\theta_{p+1}...d\theta_{p+n}. \tag{1.16}$$

Marginalising over the likehood is achieved in exactly the same way since $\mathscr{L}(\theta) = P(d|\theta)$. In the case of a likelihood with an arbitrary shape, marginalisation must be performed numerically and is well-suited to Markov Chain Monte Carlo (MCMC) methods but in the special case where it is a multivariate Gaussian, the marginalisation can be performed analytically and simply in terms of the Fisher Matrix (discussed in the next section).

Here we introduce a simple two dimensional Gaussian likehood and marginalise over a single parameter (see [20] for similar example). This lays the foundation for the general analytic solution shown in the next section. Consider the example where we have a positive semidefinite symmetric matrix,

$$F = \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix}, \tag{1.17}$$

which is used in the likelihood (given by [20]) as follows:

$$\begin{aligned}
\mathscr{L}(\theta_1) &= \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}(-\theta_{\rm A} F_{\rm AB} \theta_{\rm B})\right] d\theta_2, \\
&\propto \exp\left[\frac{F_{12}F_{21}\theta_1^2}{F_{22}} - F_{11}\theta_1^2\right], \\
&\propto \exp\left[\frac{(F_{12}F_{21} - F_{11}F_{22})}{F_{22}}\theta_1^2\right], \\
&\propto \exp\left[\frac{-\det F}{F_{22}}\theta_1^2\right], \\
&\propto \exp\left[-\frac{\theta_1^2}{\sigma_1^2}\right], \qquad (1.18)
\end{aligned}$$

where A and B run from 1 to 2 and we recognised θ_1^2/σ_1^2 as a Gaussian chi-squared, setting $\sigma_1 = \sqrt{F_{22}/(\det F)}$. We can conclude a general result for $\sigma_{\rm A}$ from the above,

$$\sigma_{\rm A} = \sqrt{(F^{-1})}_{\rm AA}. \qquad (1.19)$$

where we have used elements from:

$$F^{-1} = \frac{1}{\det F}\begin{pmatrix} F_{22} & -F_{12} \\ -F_{21} & F_{11} \end{pmatrix}. \qquad (1.20)$$

The resulting covariance matrix for two uncorrelated parameters is thus:

$$C = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}, \qquad (1.21)$$

where each $\sigma_{\rm A}$ can be expressed in terms of the matrix F. In the next section we examine the origins of F, the Fisher Matrix. It will also be shown in the proceeding section how we can generalise marginalisation of a multivariate Gaussian likelihood by subdividing the Fisher Matrix into subcomponents which can then be combined again to form an analytic solution.

1.1.8 The Fisher Framework

The Fisher Matrix translates errors on observed quantities, measured from an experiment, into constraints on parameters of interest in the underlying model. This makes it an attractive way to linearly propagate errors given multiple, correlated, measurements with many parameters [21]. As an example, consider an arbitrary function $y = f(z, \theta)$ of some parameter θ and an independent variable z. Assuming a per-

fect measurement of z, the error $\delta\theta$, for a given measured δy is, by simple calculus, $\delta\theta = \left(\frac{\partial f}{\partial \theta}\right)^{-1}\delta y$, or equivalently $\left(\delta\theta\right)^{-2} = \left(\frac{\partial f}{\partial \theta}\right)^{2}(\delta y)^{-2}$. This is perhaps the simplest example of a Fisher Matrix; with a single element, cf. Eq (1.24). Here θ represents the parameter we want to measure and f the observable quantity (e.g. $H(z)$ or $d_A(z)$, which are defined in Section 2.1 & 2.2).

If we consider a vector of parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_{\rm A}, ..)$ the Fisher Matrix formalism predicts the constraints on a vector of parameters - such as w_0 and w_a - resulting from measurements of one or more observables $\mathbf{X}^\alpha = \mathbf{X}^\alpha(\boldsymbol{\theta}, \mathbf{z})$ (such as $H(\boldsymbol{\theta}, \mathbf{z})$ or $d_A(\boldsymbol{\theta}, \mathbf{z})$), each at a range of redshifts, $\mathbf{z} = (z_1, z_2, ..., z_i, ..)$ e.g. in a BAO survey one might measure $H(\boldsymbol{\theta}, \mathbf{z})$ and $d_A(\boldsymbol{\theta}, \mathbf{z})$ at a single redshift, while a Type Ia supernova (SNIa) survey may measure $d_L(\boldsymbol{\theta}, \mathbf{z})$ at hundreds of redshifts. There are therefore three indices in general to keep track of, $({\rm A}, \alpha, i)$ corresponding to parameter, observable and redshift. The number of observables is arbitrary and combining results from independent observables is essentially trivial, hence we will often suppress observable index, $\mathbf{X}^\alpha = \mathbf{X}$ for simplicity. Boldface indicates the entire vector, either of parameters $\boldsymbol{\theta}$, observables $\mathbf{X}$ or redshifts $\mathbf{z}$.

The Fisher Matrix estimates not only the individual errors on the parameters, $\boldsymbol{\theta}$, evaluated at a given input or fiducial set of parameters $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, but also the correlations between them, leading to the characteristic Fisher error ellipsoids (ellipses if only pairs of $\theta_{\rm A}$ are considered). To make this clear, consider the likelihood, $\mathscr{L} = P(\mathbf{d}|\boldsymbol{\theta})$, for a general survey, which gives the conditional probability of observing the data $\mathbf{d} = (d_1, d_2, ..., d_i, ..)$ assuming the cosmological model $\boldsymbol{\theta}$ is correct. We can expand the likelihood around the fiducial model:

$$\begin{aligned}\ln\mathscr{L}(\boldsymbol{\theta}^* + \delta\boldsymbol{\theta}) \;=\; \ln\mathscr{L}(\boldsymbol{\theta}^*) &+ \sum_{\rm A}\left.\frac{\partial\ln\mathscr{L}(\boldsymbol{\theta})}{\partial_{\rm A}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\delta\theta_{\rm A} \\ &+ \frac{1}{2}\sum_{\rm AB}\left.\frac{\partial^2\ln\mathscr{L}(\boldsymbol{\theta})}{\partial_{\rm A}\partial_{\rm B}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\delta\theta_{\rm A}\delta\theta_{\rm B} \\ &+ \frac{1}{6}\sum_{\rm ABD}\left.\frac{\partial^3\ln\mathscr{L}(\boldsymbol{\theta})}{\partial_{\rm A}\partial_{\rm B}\partial_{\rm D}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\delta\theta_{\rm A}\delta\theta_{\rm B}\delta\theta_{\rm D} \\ &+ \frac{1}{24}\sum_{\rm ABDE}\left.\frac{\partial^4\ln\mathscr{L}(\boldsymbol{\theta})}{\partial_{\rm A}\partial_{\rm B}\partial_{\rm D}\partial_{\rm E}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\delta\theta_{\rm A}\delta\theta_{\rm B}\delta\theta_{\rm D}\delta\theta_{\rm E} + ...\,,\end{aligned} \tag{1.22}$$

where $\partial_{\rm A} \equiv \partial\theta_{\rm A}$ represents the partial derivatives with respect to the parameter $\theta_{\rm A}$. The first term in the expansion is a constant depending on the fiducial model. The fiducial model is expected (after averaging over many data realisations) to be the

point of maximum likelihood, hence the first derivative of the likelihood vanishes. The third term is the curvature matrix (Hessian) of the likelihood, and is the term used in the Fisher Matrix which is formally defined as the expectation value of the derivatives of the log of the likelihood with respect to the parameters $\boldsymbol{\theta}$, or

$$F_{\rm AB} = -\left\langle \frac{\partial^2 \ln \mathscr{L}}{\partial\theta_{\rm A}\partial\theta_{\rm B}} \right\rangle . \tag{1.23}$$

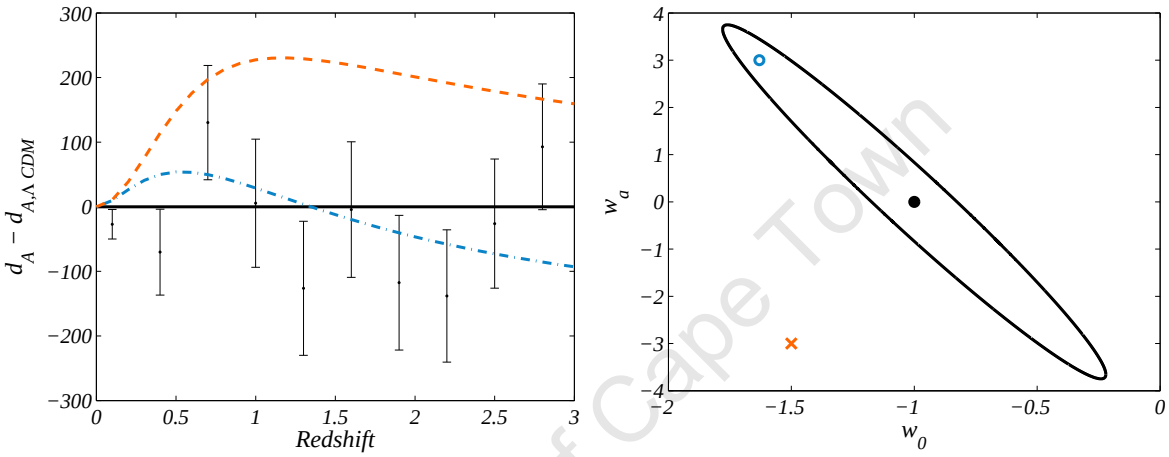


Figure 1.6: **The link between model curves, data and error ellipses [2]** – Here we show how a simulated survey measuring the angular diameter distance, $d_A(z)$, with 6% error in ten bins between $0.1 < z < 3$, assuming ΛCDM, translates into an corresponding Fisher error ellipse for the $w_0 - w_a$ plane in the right panel. The parameter values shown in left panel for the horizontal black solid line, blue dot-dashed line and orange dashed line are then depicted with their respective position in the $w_0 - w_a$ plane on the right in a corresponding colour. The parameters which are a poor fit to the data fall outside the error ellipse, e.g. orange dash line on the left and orange cross on the right, while better fitting parameters fall inside the ellipse, e.g. the horizontal solid black line where $d_A - d_{A,\Lambda\rm CDM} = 0$ on the left and the black solid dot on the right at $(w_0 = 1, w_a = 0)$.

Using the general chi-squared statistic for correlated data from Section 1.1.6, where the observable $\mathbf{X}$ is expressed in terms of the theoretical value of the observable X_i evaluated at the redshifts z_i and data for that specific observable is given by d_i. This gives a likelihood $\mathscr{L} \propto \exp(-\mathbf{\Delta}^T\mathbf{C}^{-1}\mathbf{\Delta}/2)$ where $\mathbf{\Delta} \equiv \mathbf{X} - \mathbf{d}$ as introduced previously. Substituting the above expression for the likelihood into Eq. (1.23) converts the equation from derivatives of the likelihood itself into a sum over derivatives

of the *observable* $\mathbf{X}$ with respect to the parameter $\theta_{\rm A}$:

$$\begin{aligned} F_{\rm AB} &= \frac{\partial \mathbf{X}}{\partial \theta_{\rm A}}^{\rm T} \mathbf{C}^{-1} \frac{\partial \mathbf{X}}{\partial \theta_{\rm B}} + \frac{1}{2} {\rm Tr} \left(\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta_{\rm A}} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta_{\rm B}} \right), \\ &= \sum_i \frac{1}{\sigma_i^2} \frac{\partial X}{\partial \theta_{\rm A}}(z_i) \frac{\partial X}{\partial \theta_{\rm B}}(z_i), \end{aligned} \tag{1.24}$$

where $\partial \mathbf{C}/\partial \theta_{\rm A}$ is the derivative of the data covariance matrix with respect to the parameter $\theta_{\rm A}$ which is assumed to vanish in the second equality implying that the data errors are independent of cosmological parameters. This is often the case, e.g. the errors on measurements of Type Ia supernova (SNIa) flux are independent of the dark energy parameters w_0, w_a to good accuracy. The second equality also requires that the data are uncorrelated, in which case $\mathbf{C}$ is diagonal with entries σ_i^2, with the σ_i the 1σ error on the i-th data point.

In the case where we have multiple independent measurements of different observables $\mathbf{X}^\alpha$ (e.g. $H(z)$ and $d_A(z)$), the total Fisher Matrix is just the sum of the individual Fisher matrices indexed by α. Similarly, if we have independent prior information, this is encoded in a prior matrix between the cosmological parameters. In this section we will refer to the prior on a single parameter $\theta_{\rm A}$ as ${\rm Prior}(\theta_{\rm A}) = (\Delta\theta_{\rm A})^{-2}$, where $\Delta\theta_{\rm A}$ is the uncertainty on the parameter as measured from prior surveys. In the case where the different measurements are not independent, they must be combined with the suitable data covariance matrix. The inverse of the Fisher Matrix, $F_{\rm AB}^{-1}$, provides an estimate of the error covariance matrix for the parameters $\theta_{\rm A}$, as we now expand upon.

As discussed in Section 1.1.7 marginalising over n nuisance parameters when we have a $p+n$-dimensional likelihood reduces to only a function of p key parameters:

$$\mathscr{L}(\theta_1, ...\theta_p) = \int_{-\infty}^{\infty} \mathscr{L}(\theta_1, .., \theta_p, .., \theta_{p+n}) d\theta_{p+1} ... d\theta_{p+n}. \tag{1.25}$$

If we further assume that the likelihood is a multivariate Gaussian then an analytic solution was shown to be possible in Section 1.1.7. With the concept of a Fisher Matrix now formally introduced we are able to expand on this initial illustration where we now have a general $p+n$-dimensional Fisher Matrix which can be written in terms of its sub-matrices,

$$F = \begin{pmatrix} \Theta & \mathcal{O} \\ \mathcal{O}^{\rm T} & \mathcal{N} \end{pmatrix}, \tag{1.26}$$

where Θ is the $p \times p$ sub-matrix corresponding to the parameters of interest, $\mathcal{O}$ is an $p \times n$ matrix describing the correlation between the nuisance parameters and the

parameters of interest[3] and $\mathcal{N}$ is the $n \times n$ matrix representing the nuisance parameters we wish to marginalise over. The marginalised Fisher Matrix for the parameters of interest is then given by [22]:

$$\widetilde{F} = \Theta - \mathcal{O}\mathcal{N}^{-1}\mathcal{O}^{\mathrm{T}}, \tag{1.27}$$

where the first term is the matrix of the reduced parameter space of interest, the second term encodes the effects of the marginalisation over the other nuisance parameters and ^{T} represents matrix transpose.

As we mentioned before, the inverse of the Fisher Matrix provides an estimate of the parameter covariance matrix. For an unbiased estimator (that is one whose expected value of $\boldsymbol{\theta}$ is equal to the fiducial model $\boldsymbol{\theta}^*$ assumed to be correct), and in the case where one does not marginalise over any other parameters (i.e. we consider all other parameters perfectly known), the expected error on any parameter θ_{A} satisfies the Cramér-Rao bound[4],

$$\Delta\theta_{\mathrm{A}} \geq \frac{1}{\sqrt{F_{\mathrm{AA}}}}, \tag{1.28}$$

while in the more realistic case that one wants to marginalise over all the other parameters in the problem, the bound becomes,

$$\Delta\theta_{\mathrm{A}} \geq \sqrt{(\mathbf{F}^{-1})_{\mathrm{AA}}}, \tag{1.29}$$

i.e. one first inverts the Fisher Matrix, then takes the AA component of the resulting matrix. One can show that the latter is always greater than or equal to the former, i.e. marginalisation cannot decrease the error on a parameter, and only has no effect if all other parameters are completely uncorrelated from the parameter of interest. Note that in the case where the likelihood is exactly Gaussian in the parameters, the Cramér-Rao bound becomes an equality and not just a lower bound.

Since Fisher Matrix analysis assumes the likelihood is a multivariate Gaussian, contours of constant probability are ellipsoids within the Fisher formalism. These ellipsoids (ellipses for two parameters) are given by solving the equation,

$$\boldsymbol{\Delta\theta}^{\mathrm{T}}\widetilde{F}\boldsymbol{\Delta\theta} = \beta, \tag{1.30}$$

where $\boldsymbol{\Delta\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$ is the parameter vector around the fiducial model, $\boldsymbol{\theta}^*$, and β is a constant determined by the desired confidence level and the number of parameters. For two parameters, the 1 and $2-\sigma$ contour levels correspond to $\beta = 2.31$

[3] If $\mathcal{O} = 0$ then the nuisance parameters have no impact on Θ.

[4] For a proof see e.g. p. 426 in [18]

and 6.17 respectively [22,23]. In Chapter 5-6 the Fisher4Cast software is introduced and its applications highlighted. This is a toolbox, written in Matlab, which aims to automate the Fisher Matrix calculations in a completely general way, while adding numerous additional cosmological specific features to help with survey forcasting in the Graphical User Interface (GUI). The Fisher4Cast GUI allows plotting of both 1- and 2-dimensional contours and hence always marginalises the full 5-dimensional Fisher Matrix to achieve this. Marginalisation over some or all of the other parameters can be effectively switched off by making the corresponding diagonal elements of the prior matrix very large. The Fisher Matrix and the corresponding ellipses provide the Gaussian estimate for how well the parameters of the model will be constrained by a given experiment assuming the true model is that at which the Fisher Matrix was evaluated (e.g. ΛCDM). This is illustrated in Figure 1.6, which shows the $1-\sigma$ error ellipse around the fiducial ΛCDM model with the coefficients in the Chevallier-Polarski-Linder (CPL) [8,9] parameterisation $(w_0, w_a) = (-1, 0)$, for a survey consisting of measurements of the angular diameter distance between redshifts of 0.1 and 3. Values of w_0, w_a inside this ellipse will have expected likelihoods that differ from the fiducial model by less than $1-\sigma$. The Fisher Matrix allows us to estimate which sets of parameter values we will be able to rule out at a given significance level if the fiducial cosmological model is correct.

1.2 Optimization

A general problem that is common to a wide spectrum of endeavours is using a limited resource in the most effective way. This usually means that a maximum return or minimum cost is desired for a given situation and its associated constraints. A simple example would be finding the shortest distance between two points or possibly the fastest route between two points, while the limiting factors would be the top speed of the transport being using and the roads or paths available between the two points. This is the domain of optimization and it is used in a myriad of applications from industry to finance.

This also extends to cosmology where optimization has become more common place [24, 25]. Selecting an optimal subset of targets from a large catalog would be a simple example of how optimization can have a very real impact on everyday astronomy and cosmology, as shown in [26].

Optimization can be defined as the process of maximising or minimising a function in a given range. The goal of optimization is to solve these problems either

analytically, where possible, or else through heuristic algorithms, when not possible.

Here we define a global minimum for a function as being:

$$\begin{aligned}&\text{Given } f : \mathbb{R}^n \to \mathbb{R},\\ &\text{find } x^* \in \mathbb{R}^n \text{ such that } f(x^*) \leq f(x), \forall x \in \mathbb{R}^n.\end{aligned}$$

There are three characteristic elements to an optimization problem:

1. The objective function or figure of merit (FoM) represents the function that is to be optimized.

2. A set of variables which the objective function is dependent on, e.g. parameters of a model, x.

3. A range of constraints that are applicable, e.g. non-negative time.

Constraints, however, are not always provided or required and these unconstrained problems are an important field of study.

In Figure 1.7 we illustrate the difference between a local and global minimum. A local minimum, x_B^*, is the minimum in a given region, B (which is shaded), while the global minimum, x^*, is the true minimum in the entire search space considered. These two minima can be placed in a more rigorous context where, x_B^*, is defined as,

$$f(x_B^*) \leq f(x), \forall x \in B,$$

and $B \subset S \subseteq \mathbb{R}$ where S represents the search space. While the global minimum is defined as,

$$f(x^*) \leq f(x), \forall x \in S.$$

It is customary in the unconstrained case to use $S = \mathbb{R}^n$, where n is the dimension of x.

Optimizing problems can in the simplest cases be solved analytically but often more difficult problems have a level of complexity that makes an analytic solution impossible. In the case of the more simple problems deterministic algorithms are used which often employ numerical methods. An example of some of the more popular deterministic algorithms are the Newton-Raphson method, Steepest Decent [27], Scaled Conjugate Gradient [28], and Quasi-Newton methods [29]. In contrast the non-deterministic methods used on challenging problems requires a more sophisti-

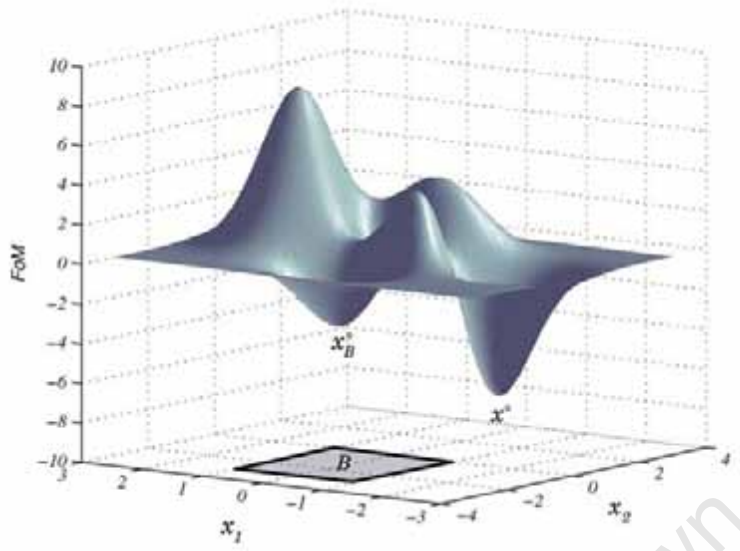


Figure 1.7: **Global and Local Minima** – Here a function[5] is shown to help distinguish a local minimum, x_B^* in the shaded region B, from the global minmum, x^* across the whole space.

cated approach and for this purpose heuristic algorithms are employed.

A heuristic algorithm abandons the notion of an exact solution and rather employs a methodology of trial and error to reach an optimal solution. We can understand this in simple terms by taking the example of packing a suitcase. It would be very difficult to formulate an analytic solution of how we can pack the content of a bag for all varieties of content and containers in the most compact way. A strategy that is general and reliably good is to first pack the larger things and progressively pack the smaller content on top of that, into the bag. This is a heuristic procedure.

1.2.1 Standard Algorithms

There is no standard recipe for generating optimizing algorithms but nature is often used as a good starting point since it has developed dynamics that favour finding the best outcomes and does so in a diverse range of ways. A prime example of this would be the evolutionary strategy employed by a Genetic Algorithm (GA) [30–33] which uses the principle of cross breeding and mutation of genes to formulate an optimal solution. Other optimization algorithms often used include Ant Colony Optimization (ACO) [34,35], Artificial Neural Networks (ANN) [27], Simulated Annealing (SA) [36]

[5] $f(x_1, x_2) = 3(1-x_2)^2 \exp[-x_2^2-(x_1+1)^2] - 10(\frac{x_2}{5}-x_2^3-x_1^5)\exp[-x_2^2-x_1^2] - \frac{1}{3}\exp[-(x_2+1)^2-x_1^2]$

and Particle Swarm Optimization (PSO) [37, 38].

Each of these optimization techniques has their own strengths and weaknesses which makes them better suited to particular applications. GA and ACO are powerful techniques but they can be complicated to implement and generally require a great deal of expertise to successfully be used. In the remainder of this introduction we will be concentrating on SA and PSO which are more simple to apply to a diverse set of problems and become the focus of Chapter 3. We next look at SA in more detail which has a very basic algorithm making it easy to code along with being particularly well suited for a broad range of problems.

Simulated Annealing

Simulated Annealing (SA) [36] is an optimization algorithm which derives its name from the process of annealing in Metallurgy where a metal is heated to a high temperature and then cooled in a slow and systematic way. This allows the atomic structure of the metal to become unbonded and then explore an assortment of random configurations while the temperature is high. This facilitates the exploration of lower energy configurations and the system will favour and eventually settle in these stable states as the temperature is decreased.

This will generally be the most stable bonded state which in turn will result in the metal's strength being greatly increased, provided that the process was started at a sufficiently high temperature and the cooling is done slowly enough. Figure 1.8 shows an example where annealing is used as a technique to promote the production of single crystal silver particles, which in turn are more stable than the silver nanocrystals deposited in the beginning [4].

The heuristic of a SA optimization algorithm [36] is thus exactly analogous to this process and relies on the same principle that the system will converge on a minimum or maximum FoM, where the FoM plays the same role as the bonding energy in the case of annealing. There is an inherent danger in SA, much the same as in annealing, that the system can converge to and get trapped in local minima. This problem typically arises when rapid cooling takes place which doesn't give the system sufficient opportunity to exhaust all the energy configuration before converging on the best option. One effective way of overcoming this is to allow for a probability of accepting a worse position of a higher FoM. This allows the algorithm to more broadly explore varying paths and not just limit it search to what appears to be a better location in the state space. Metropolis and Hastings sampling [39, 40] is a classic implementation which substantially averts the above shortfall and is thus used

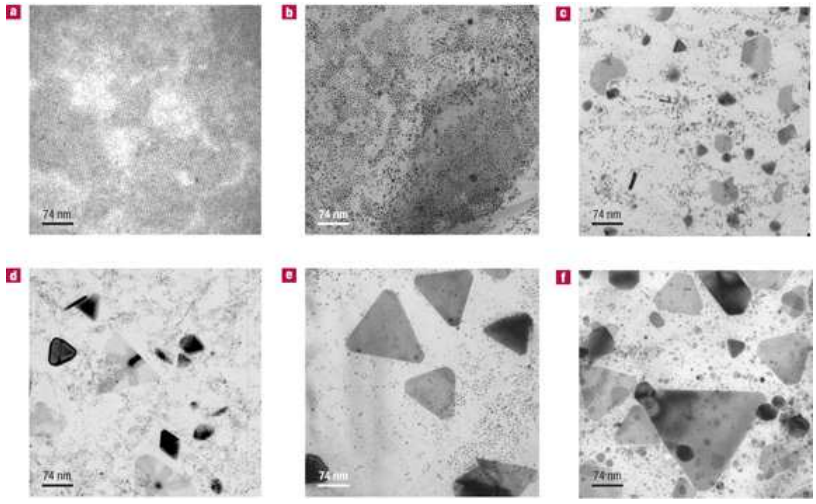


Figure 1.8: **Example of Annealing [4]** – This figure shows the emergence of silver triangular crystals through the use of annealing. This becomes more pronounced with an increases in size as the period of annealing is extended. Results after an annealing time of 3 hours (a), 1 day (b), 2 days (c), 3 days (d) 6 days (e) and 8 days (f).

in standard SA[6].

We follow a notational convention where a position in the parameter space at a given iteration, j, is represented by $x(j)$ where $j \in \{1, 2, \ldots, n\}$ and n is the total number of iterations. The FoM is returned by the position in the parameter space and a shorthand is used where FoM(j) represents the FoM of a given position, $x(j)$, after j iterations.

Next we run through an outline of a standard SA algorithm using MH sampling with the help of a flowchart, shown in Figure 1.9, and discuss the component of the flowchart in the proceeding pseudo code. The pseudo code assumes throughout that the SA algorithm is seeking the minimum FoM.

Pseudo Code

- Initialize: The first step is to set the initial position and conditions for convergence. Additional values to set to zero would be the current best position and the number of iterations done. An initial run of a certain number of iterations may also be done before the recording of data begins. This is referred to as a "burn in" period and is used to help eliminate bad statistics that may emerge from the initial transient. The necessity for a burn in period is a point of contention and is discussed futher in [41].

[6] A step-by-step guide of MH sampling is explained in the pseudo code for a standard SA algorithm

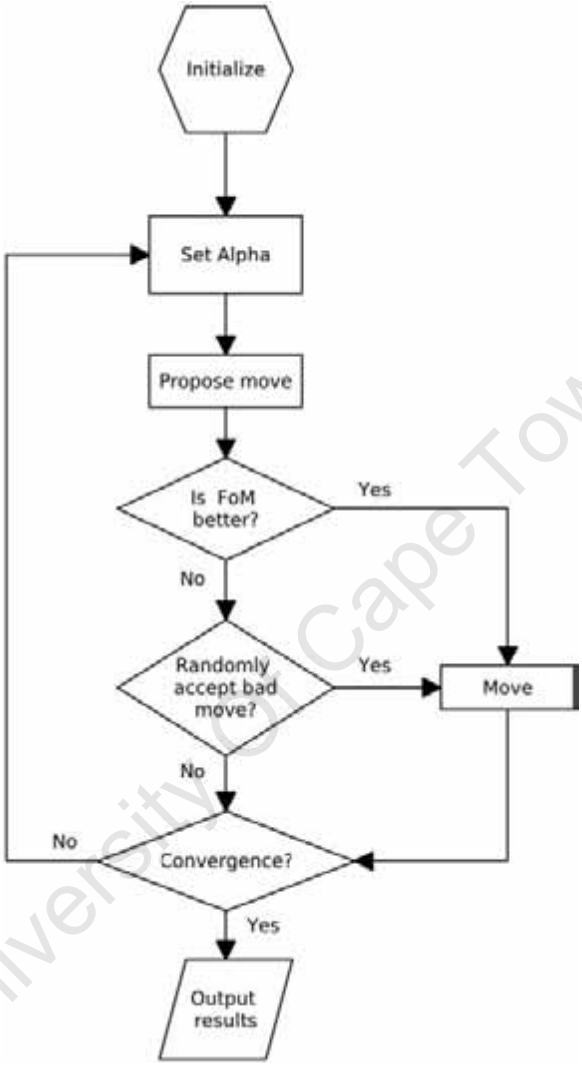


Figure 1.9: **Flowchart for standard Simulated Annealing Algorithm** – This flowchart outlines the required components and sequence to build a standard SA algorithm. The accompanying pseudo code in the text describes each element listed in the flowchart. This flowchart also serves as a guideline for the MCMC algorithm where α would be set to a constant value.

- Set α: At this step we increase α by a set amount as given by the cooling schedule:

$$\alpha(j) = 1/T(j) = \log(j+1)/T_0, \tag{1.31}$$

where T_0 is a factor (often referred to as the initial temperature) which sets the rate at which the acceptance probability decreases as the number of iterations increase. Thus bad steps are less likely to be accepted towards the end of the run of the algorithm.

- Propose move: Next a new position is proposed:

$$x(j+1) = x(j) + \sigma, \tag{1.32}$$

where σ is chosen from a proposal distribution which is usually taken to be a Gaussian. The FoM, of the desired object function, is calculated at the new position and the difference in the proposed FoM$(j+1)$ and current FoM(j) is calculated,

$$\Delta\text{FoM} = \text{FoM}(j+1) - \text{FoM}(j). \tag{1.33}$$

- Is FoM better?:

  - If the newly considered position has a lower FoM than the previous position, Δ FoM< 0, then accept the move to the new position.

- Randomly accept bad move?:

  - If the newly proposed position has a higher FoM than the current position a move wouldn't normally be allowed. The Metropolis Hastings (MH) algorithm does however allow these bad moves with a certain probability. This helps to ensure that the system doesn't converge in local minima and the algorithm can "climb" its way out of bad positions to better explore the entire space for a global minimum. The probability is given by:

$$P(x(j)|x(j+1)) = \min\{1, \exp[-\alpha\Delta\text{FoM}]\}. \tag{1.34}$$

Thus if the uniform random number generated, between [0,1], is less than $\exp[\alpha\Delta\text{FoM}]$ then the bad move is accepted.

- Record data: This is an optional step not shown in the flowchart of Figure 1.9 but can be used to record data that can be used for postprocessing analysis. Data that can be recorded includes the lowest FoM found at each step and its

position in the parameter space along with the α value at each iteration. The number of accepted and rejected steps can also be recorded to analyse what the acceptance ratio of proposed steps was.

- Convergence?: Here we consider the convergence criteria which needs to be met before the while loop ends. There are a range of criteria that can be used for convergence. The simplest is to set a fixed number of iterations or a fixed length of time that the code runs for. More sophisticated methods would set a threshold of accuracy for the FoM or cease when the FoM doesn't change significantly over a set number of iterations (FoM stagnates). If the while loop is not ended then the algorithm loops back to the second step.

- Output results: This is done once the loop has completed and the statistics of the run including the best position and value can be displayed, plotted or written to file.

The well known Markov Chain Monte Carlo (MCMC) algorithm, which is often used in combination with the MH sampling method, is equivalent to the SA algorithm where $\alpha = 1$. This implies that the acceptance probability, Eq. (1.34), does not change as a function of iteration.

The overriding factors that contribute to SA being very popular are its basic code structure which allows for easy implementation to a variety of problems while being robust and reliably finding the global minimum, even when dealing with a complex parameter spaces. Although there exist variants on the standard SA there are none that address the idea of sharing information between multiple chains. This is further explored in Chapter 3. An optimization algorithm that successfully employs a strategy of sharing information between multiple particles is PSO, which we next look at.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) first introduced by [37], was inspired by the sociological behaviour associated with birds flocking and the dynamics at play in establishing the flock as a collective entity. In nature swarming, establishing a herd or a flock has many evolutionary advantages since it aids survival and distributing the responsibility of finding a food source. This underlies the optimization principle at work which favours the collective resources of the group as opposed to that of the individual alone.

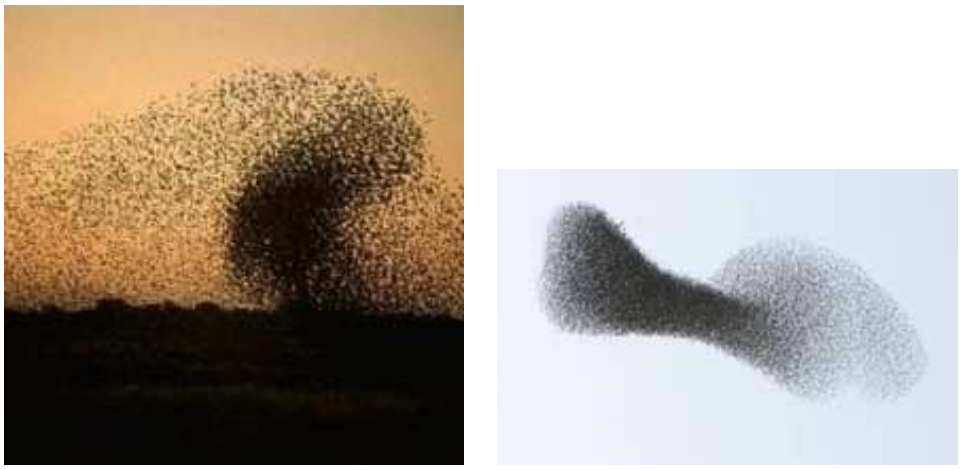


Figure 1.10: **Swarm in Nature** – Illustration of a flock of birds in the left panel (Credit[7]) while a simulated swarm of particles is shown in the right panel (Credit[8]).

We can build a simple algorithm to mimic the nature of a swarm by assigning an object function to each of the particles composing the swarm. The object function is referred to as the figure of merit (FoM) throughout this thesis. The position of a particle is specified by the vector $\vec{x}_i(j)$ and the velocity vector of the same particle is given by $\vec{v}_i(j)$ while its best position up until that iteration is given by $y_i(j)$. Each particle is identified by a subscript $i \in \{1, 2, \ldots, s\}$ where s is the total number of particles in the swarm. As before the iteration is given by $j \in \{1, 2, \ldots, n\}$ where n is the total number of iterations.

This is governed by a simple model of individual birds within a flock (or particles in a swarm) and how the direction of the individual particle is determined by the knowledge of its own previous best position and the best position of a neighbouring bird (global influence). These heuristics combine in such a way as to allow the flock to swarm and navigate its way through the parameter space until it settles on an optimum location. The premise is that the collective direction of the swarm is dependent on the findings of the best prospects within the swarm and each individual's best previous position. This means that the individuals or particles from within the swarm share information between each other and the best location is accelerated towards while being tempered by the individuals own previous best position prior to that information being known.

The highest FoM of any particle within a neighbourhood is found and then all the particles become aware of that position due to the sharing of information amongst the

[7] www.thomevincent.com/blog/index.php?2007/12
[8] flowingdata.com/2007/07/04/social-data-analysis-by-the-swarm

swarm. All the particles then jointly accelerate towards both the best position of the highest FoM of a given neighbourhood, $\hat{y}(j)$, and their own previous best position, $y_i(j)$. This process can be quantified in the velocity equation for each iteration,

$$\vec{v}_{i+1}(j) = \eta\vec{v}_i(j) + c_1 r_i^1(j)\underbrace{[y_i(j) - \vec{x}_i(j)]}_{\text{personal influence}} + c_2 r_i^2(j)\underbrace{[\hat{y}(j) - \vec{x}_i(j)]}_{\text{global influence}} \tag{1.35}$$

where c_1 and c_2 are called the acceleration coefficients and the FoM at the current position is given by $x(j)$. The stochastic component of the equation is composed of the two random numbers, r_i^1 and r_i^2, which allow each particle, i, to be weighted randomly in favour of either the *personal* or *global* influence. The personal and global influences are also known as the cognitive and social interaction components. In order to then update the position of each of the particle, i, we simply add the velocity of that particle to its current position,

$$\vec{x}_i(j+1) = \vec{x}_i(j) + \vec{v}_i(j+1). \tag{1.36}$$

There are two predominant forms of PSO, global best and local best models. The distinguishing factor between these two models is the range of particles considered in order to locate the best FoM. In the case of the global best models the entire swarm is considered while in local best models only a defined neighbourhood is considered. The global best model has the advantage of being able to converge very quickly on a solution but this is done at the cost of generality. This makes it more susceptible to converging on local minima while the local best model doesn't have the same speed but it is able to be broadly applied to a function and converging to the true global minimum with a greater level of confidence.

Next we run through an outline of a typical PSO algorithm with the help of a flowchart, shown in Figure 1.2.1, and discuss the component of the flowchart in the proceeding pseudo code.

Pseudo code

- Initialize: Here we initialize the position of the swarm. We also specify the number of particles in the swarm along with the convergence criteria. This could include a set number of iterations or a threshold FoM value.

- While loop: We test to see if the convergence criteria has been met and whether the loop should terminate.

- Calculate FoM for all particles: We next calculate the FoM for each of the

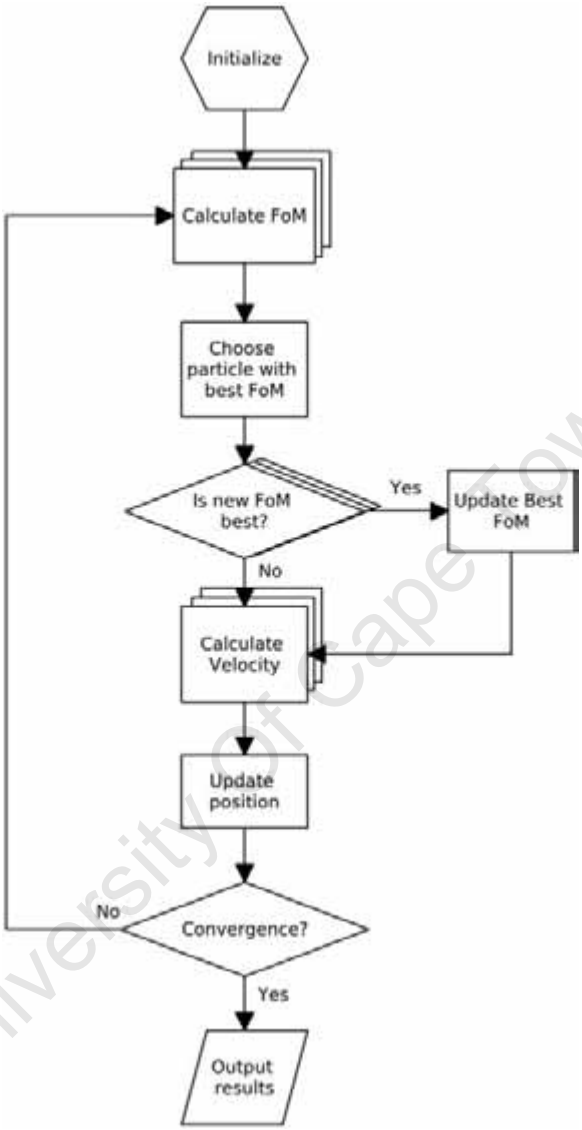


Figure 1.11: **Flowchart for a Particle Swarm Optimization Algorithm** – This flowchart outlines the required components and sequence to build a PSO algorithm. The accompanying pseudo code in the text describes each element listed in the flowchart. The multiple layered functions in the flowchart indicate those specific subcomponent of the algorithm that are applied to each particle, i, individually.

particles in their current positions.

- Choose particle with best FoM: We next determine the best FoM for all particles in a defined neighbourhood (could be small or be the entire swarm).

$$\max\{\vec{x}_i(j)\} \tag{1.37}$$

where $i = 1, 2, \ldots, r$ and r defines the size of the neighbourhood.

- If best FoM:

  - If the newly evaluated FoM is better than the previous best global FoM, then replace it as the new best global FoM.

- Calculate velocity: Next calculate the velocity, using Eq. (1.35), for each of the particles using the best FoM of the individual particles up until that iteration, $y_i(j)$, and the current best global FoM, $y(\hat{j})$.

- Update position: We now update the position, using Eq. (1.36), of all the particles using the newly calculated velocity and adding it to the particles current position.

- End while loop: We now loop back to the second step of checking to see if the convergence criteria has been met yet. If not we continue the loop and follow the same procedure else we proceed to the output of the data.

- Output results: This is done once the loop has completed and the statistics of the run including the best position and FoM can be displayed, plotted or written to file.

In the simplest version of PSO only a global best is considered and there is no accounting for each particle memorizing its previous best FoM. The version of PSO described above is thus slightly more complex than the very simplest incarnation of PSO but most variants build on a version of this form. It thus helps to show the interplay between local and global influences in the PSO heuristic.

The strength of PSO lies in the fact that it can share information between the particles. This can lead to quicker convergence times in complex problems. The disadvantages are that it is a more complex algorithm to implement and can have a longer run time than for example SA. The intriguing question that arises is how we can combine the strengths of both PSO and SA to provide a better hybrid optimization strategy. This is addressed in Chapter 3 where we also look at how we can compare

differing optimization algorithms on an equal footing to identify the most promising candidates.

1.3 Conclusion

This chapter introduced basic statistical elements culminating in the Fisher Matrix formalism. This becomes an important aspect of Chapter 5 and 6 where a software suite able to do Fisher Matrix calculations is presented and its capabilities explored. In addition the concept of optimization and some standard heuristic algorithms were introduced and reviewed. This forms a basis for Chapter 3 which discusses a new optimization algorithm and compares its performance with standard optimization algorithms.

Chapter 2

Introduction to Cosmology

"*In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move.*"
Douglas Adams

Cosmology is a vast field and usually requires a broad exposition to grasp its complexity. Fortunately there are only certain fundamental aspects of cosmology that pertain to the upcoming chapters and can be summarised in a self contained introduction presented here. For a more thorough review see [20, 42–44].

2.1 Einstein Equations

The Einstein equations provide a link between the geometry of spacetime, $G_{\mu\nu}$, and the stress-energy tensor, $T_{\mu\nu}$, of matter:

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} \mathcal{R} = 8\pi G T_{\mu\nu}. \tag{2.1}$$

Here $G_{\mu\nu}$ is the Einstein tensor and $R_{\mu\nu}$ is the Ricci tensor [43] which is dependent on the metric, $g_{\mu\nu}$, and its derivatives. While R is the Ricci scalar formed from the contraction of the Ricci tensor $R \equiv g^{\mu\nu} R_{\mu\nu}$ while G is Newtons' constant. The key geometric building block is the metric:

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu, \tag{2.2}$$

which describes the curvature of any space. The Einstein equations, in Eq.(2.1), are supplemented by energy momentum conservation, or equivalently, the Bianchi

identities, which state that

$$G^{\mu\nu}{}_{;\nu} = 0 = T^{\mu\nu}{}_{;\nu}, \tag{2.3}$$

where the subscript ; is shorthand for the covariant derivative [43]. The resulting set of ten nonlinear coupled partial differential equations are challenging to solve and exact solutions are only known in special cases, typically with high symmetry. Fortunately the universe is close to isotropic in both galaxy counts, see Figure 2.1, and the Cosmic Microwave Background (CMB), see Figure 2.2. An additional simplification is given by the Copernican principle which states that the universe is homogeneous or invariant under spatial translations. While homogeneity has not yet been proved convincingly [45, 46] it is the simplest starting point and leads to predictions that are in good agreement with observations. Combining the simplification of isotropy and homogeneity leads to the Friedmann-Lemaître-Robertson-Walker (FLRW) spacetimes [20, 42–44], which is the most general constant curvature spacetime:

$$ds^2 = -dt^2 + a(t)^2 \left[\frac{dr^2}{1-kr^2} + r^2 d\Omega^2\right], \tag{2.4}$$

where $d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2$ and $k = -1, 0, +1$ represent the three cases of spatial curvatures (open, flat, closed). The signature in Eq. (2.4) is $[-1, 1, 1, 1]$. In Figure 2.3 an example of an exotic space with constant negative curvature is shown.

The radiation from a distant astronomical object, s, moves to progressively longer wavelengths as it recedes from an observer, o, due to the background expansion of the universe. This measure of displacement from an observer is called the redshift, z, and is defined as:

$$z \equiv \frac{\lambda_o}{\lambda_s} - 1, \tag{2.5}$$

which is related to the scale factor, $a(t)$, as,

$$\frac{a}{a_0} = \frac{1}{1+z}, \tag{2.6}$$

where a_0 is the value of the scale factor today. The energy-momentum tensor of a perfect fluid is given by,

$$T_{\mu\nu} = (\rho + p)U_\mu U_\nu + p g_{\mu\nu}, \tag{2.7}$$

where U^μ is the fluid four vector. This is a standard way in which to model the energy-momentum sources and assumes an isotropic pressure, p, and energy density, ρ.

Using Eq. (2.7) with the metric Eq. (2.4) provides the FLRW solution for the

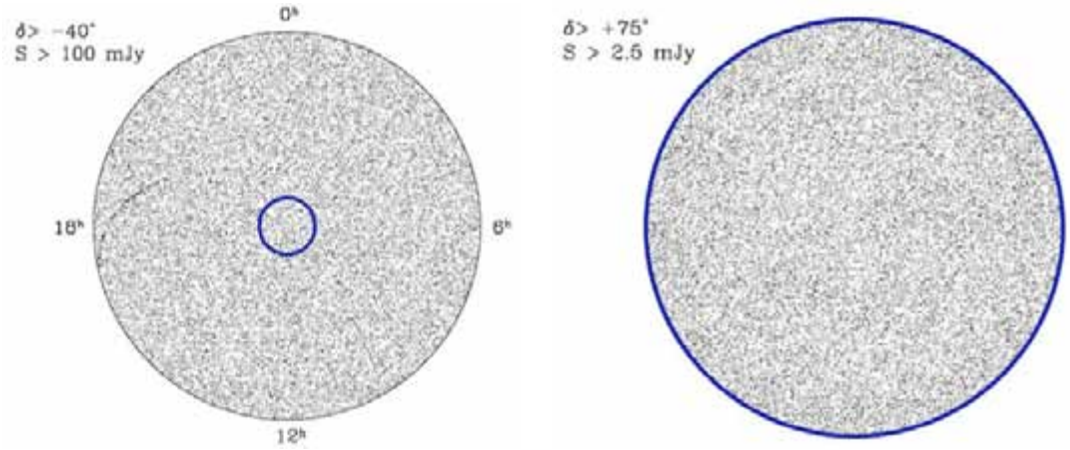


Figure 2.1: **Isotropy in NVSS Data** – The left panel shows the NRAO VLA Sky Survey (NVSS) which is a whole sky survey north of declination $\delta > -40°$ with sources above 100mJy. The right panel shows a "zoomed in" view (blue region in left panel) above $\delta > 75°$ with sources greater than 2.5mJy. The result shows the discrete sources across the sky to be extremely isotropic. Note that this figure is good evidence for isotropy, but is not strictly a proof. It would in principle be possible to take distributions of galaxies at different redshifts which are very anisotropic taken separately but which, when combined, give an isotropic distribution. This is very fine-tuned however, especially as the isotropy is preserved at different flux limits (Credit[1]).

Einstein equations producing the Friedmann equation by solving the time-time component, G_{00},

$$H^2 \equiv \left(\frac{\dot{a}}{a}\right) = \frac{8\pi G}{3}\rho - \frac{k}{a^2}, \tag{2.8}$$

while the combination of both the time-time components, G_{00}, and the space-space component, G_{ii}, produces the acceleration equation,

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}(\rho + 3p). \tag{2.9}$$

as described in [20] on page 151. Energy momentum conservation dictates that,

$$\dot{\rho} + 3H(\rho + p) = 0. \tag{2.10}$$

The density required to ensure that Eq (2.8) produces a zero spatial curvature (k=0) is defined as the critical density:

$$\rho_{\rm crit}(z) \equiv \frac{3H(z)^2}{8\pi G}. \tag{2.11}$$

[1] www.cv.nrao.edu/course/astr534/ExtraGalactic.html.

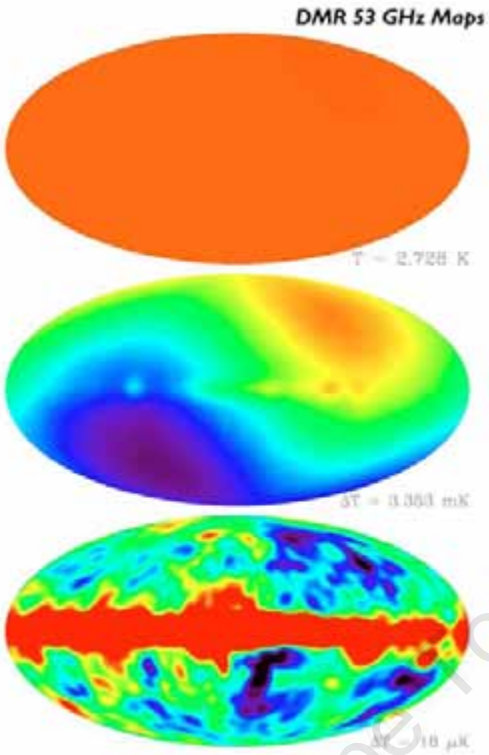


Figure 2.2: **COBE 4 Year DRM Results** – Each of these three panels shows the variation of temperature for the CMB across the sky. The top panel shows a temperature scale between 0 Kelvin (blue) and 4 Kelvin (red). This shows a completely uniform temperature across the sky, thus substantiating the isotropic assumption. The middle panel shows a scale which ranges between 2.721 (blue) and 2.729 Kelvin (red). This produces the dipole anisotropy that results from the motion around the Sun relative to the rest frame of the CMB. The bottom panel shows the anisotropy map with the dipole removed. The plane of the Milky Way still however remains and runs horizontally across the center of the image (Credit[2]).

The Universe is composed of differing components of energy density, which are diluted as the Universe expands depending on their equations of state parameter w. Each of the corresponding energy densities can be written in terms of the critical density today,

$$\Omega_r = \frac{\rho_r(z=0)}{\rho_{crit}(z=0)},\ \Omega_m = \frac{\rho_m(z=0)}{\rho_{crit}(z=0)},\ \Omega_{\rm DE} = \frac{\rho_{\rm DE}(z=0)}{\rho_{crit}(z=0)}, \tag{2.12}$$

where each of the subscripts r, m and DE represent radiation, matter and dark energy respectively. The later is responsible for fuelling the accelerated expansion of the Universe and is often assumed to be a cosmological constant, as described in Section 2.5.1. The common form of the Hubble function can be produced by combining

[2]map.gsfc.nasa.gov.

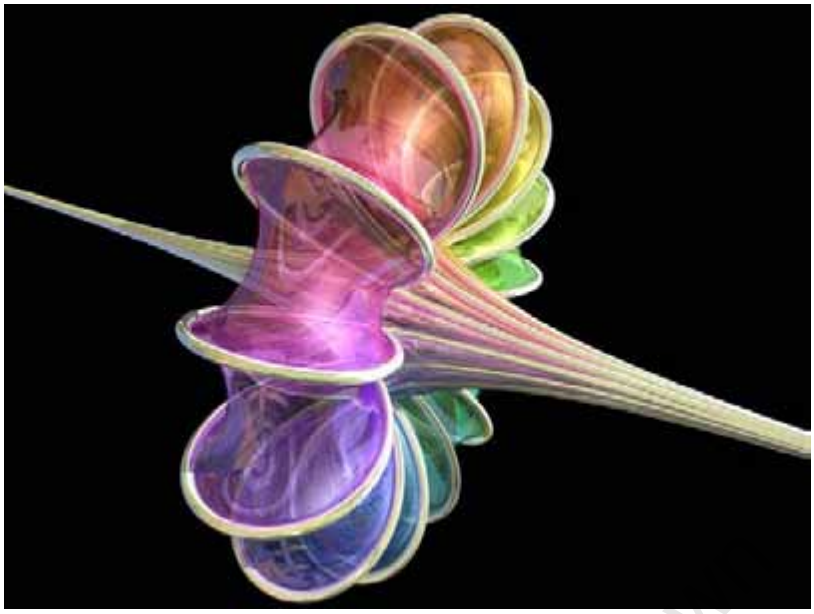


Figure 2.3: **Negative Curvature** – A unique illustration (Credit[3]) of a pseudosphere which is a hyperbolic surface with constant negative curvature not often seen.

the above two equations (Eq (2.11) & Eq (2.12)):

$$H(z) = H_0 E(z) \equiv H_0 \sqrt{\Omega_m (1+z)^3 + \Omega_k (1+z)^2 + (1 - \Omega_m - \Omega_k) f(z, w_0, w_a)}, \tag{2.13}$$

where H_0 is the Hubble constant and $f(z)$ is defined as,

$$f(z) \equiv \exp\left[3 \int_0^z \frac{1 + w(z')}{1 + z'} dz'\right], \tag{2.14}$$

where $w(z)$ is the equation of state parameter is defined as $w(z) \equiv \rho/p$. The total energy density is composed of the sum of its four sub-components,

$$\Omega_r + \Omega_m + \Omega_{\rm DE} + \Omega_k = 1, \tag{2.15}$$

where the subscripts r, m, DE and k are representative of radiation, matter, dark energy and curvature. There have been three distinct eras of dominance spanning the evolution of the universe. Determining dominance is a measure of which energy density contributes the most to the total energy density. Each of the contributing components of energy density has diluted as the universe has expanded and can be

[3] bugman123.com/Math/Math.html.

characterised by the evolution of the densities,

$$\rho_i(a) = \frac{H_0 \Omega_i(a)}{a^{3(1+w_i)}}, \tag{2.16}$$

where i applies for radiation, matter and dark energy. The energy density for curvature can be obtained via the constraint that the sum of the other components is unity, as shown in Eq. (2.15). Each of the energy densities have been experimentally measured by a range of cosmological probes and the results are listed in Table 2.1.

The energy components of radiation is often neglected due to its very small contribution to the total energy density ($\sim 10^4$ smaller than matter and dark energy). It accounts for electromagnetic radiation and relativistic matter (e.g. neutrinos) with $w = 1/3$ resulting in a^4 for Eq. (2.16).

The energy density of matter has two sub-components composed of baryonic, Ω_b, and non-baryonic or "dark matter", $\Omega_{\rm DM}$, the latter being inferred from observations. When cold dark matter (non-relativistic) is considered the resulting pressure for both components of matter is zero which leads to an equation of state parameter of $w = 0$ producing a^3 for Eq. (2.16).

Finally dark energy, the force that drives the acceleration of the universe, is often assumed to be equivalent to the cosmological constant Λ for simplicity. This is discussed in more detail in Section 2.5.1. An equation of state parameter parameter in this case is given by $w = -1$ resulting in $\Omega_{\rm DE}(a) = \Omega_{\rm DE}(a = 1)$ from Eq. (2.16).

Measured Cosmological Parameters				
Tests of Geometry				
Probe	Reference	Ω_b	$\Omega_{\rm DM}$	$\Omega_{\rm DE}$
CMB	[12]	0.0441 ± 0.003	0.214 ± 0.027	0.742 ± 0.030
CMB + BAO + SNe	[12]	0.0462 ± 0.002	0.233 ± 0.013	0.721 ± 0.015
SNe Ia	[47]	0.31 ± 0.21		0.80 ± 0.31
SNe Ia + BAO	[47]	0.27 ± 0.02		0.75 ± 0.08
BAO + SNe + CMB	[48]	0.252 ± 0.027		0.743 ± 0.047
Tests of the Growth of Structure				
Clusters + CMB	[49]	0.265 ± 0.016		$1 - \Omega_m$

Table 2.1: **Measured Parameters by Combining Cosmological Probes** – The values of cosmological parameters as measured by individual and combined probes.

2.2 Distances

In an expanding universe we need a more general notion of distance than that offered by Euclidean geometry. A comoving distance can be used as a means to keep the measured distance between points invariant under an expanding background. We define the line-of-sight comoving distance, $D_{\rm C}$, as [50],

$$D_{\rm C}(z) = \frac{c}{H_0}\chi(z) \equiv \frac{c}{H_0}\int_0^z \frac{dz'}{E(z')}, \tag{2.17}$$

while the transverse comoving distance is related to $D_{\rm C}$ as [50]:

$$D_{\rm M}(z) = \frac{c}{\sqrt{|\Omega_k|}}\text{sinn}\left[\sqrt{|\Omega_k|}\ \chi(z)\right], \tag{2.18}$$

where "sinn(x)" denotes sinh(x) when $\Omega_k > 0$, sin(x) when $\Omega_k < 0$, and x when $\Omega_k = 0$. We next define standard measures of distance which can be extracted from observational quantities.

The *angular diameter distance*, $d_A(z)$, is defined as the ratio of the object's physical transverse size to its angular size (in radians). We are thus able to convert the angular separations between two objects, from an observers viewpoint, into the proper separation at the source. The angular diameter distance is related to the transverse comoving distance by,

$$d_A = \frac{D_{\rm M}}{1+z}. \tag{2.19}$$

The flux received from a distant source decreases spherically symmetrically the further away it is. If we know the intrinsic luminosity we are able to calculate how far away the object is. This is what makes standard candles[4] such as Supernova, discussed in Section 2.4, so important in determining astronomical distances to source galaxies. The ratio of the observed bolometric flux, F, with a bolometric luminosity, L, defines the *luminosity distance*:

$$d_L = \frac{L}{4\pi F}. \tag{2.20}$$

A useful expression that relates all three of the above distance measures in terms of each other, is given by:

$$d_L = (1+z)D_M = (1+z)^2 d_A, \tag{2.21}$$

where the angular diameter distance is multiplied by one factor of $(1+z)$ from the loss of energy of the photon being stretched by the expanding universe, and an additional

[4]Objects with a well known absolute magnitude are called standard candles.

factor of $(1 + z)$ due to time dilation.

A common form of expressing distances in astronomy is the *distance modulus*, μ. This is given in terms of the apparent magnitude, m, (the apparent brightness of an object at its true distance from the observer) minus the absolute magnitude, M (the brightness of that object if we were 10 parsec away) [50]:

$$\mu = m - M = 5 \log_{10} \left(\frac{d_L}{10\text{pc}} \right), \tag{2.22}$$

where d_L is measured in parsec.

2.3 Cosmic Microwave Background

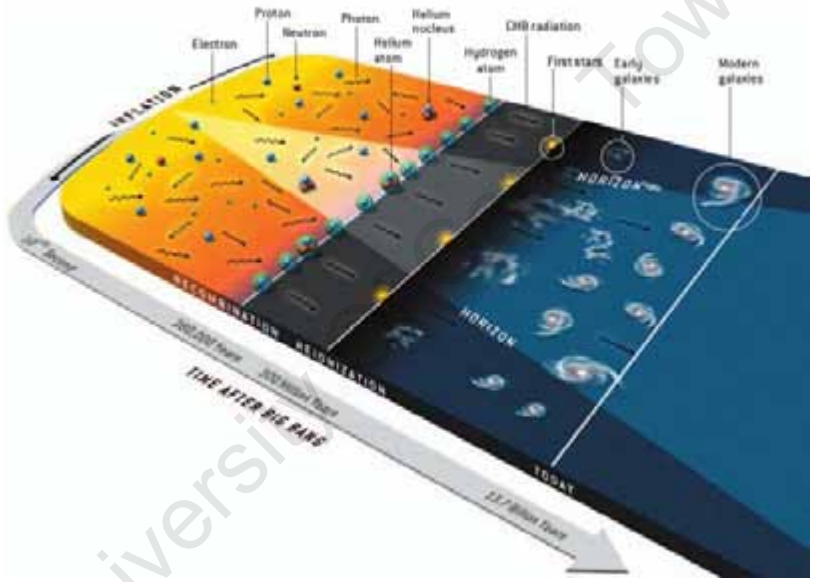


Figure 2.4: **Timeline of the Universe [5]** – As inflation expanded the universe, the plasma of photons and charged particles grew far beyond the horizon. During the recombination period about 380,000 years later, the first atoms formed and the Cosmic Microwave Background (CMB) radiation was emitted. After another 300 million years, radiation from the first stars reionized most of the hydrogen and helium. Note that in most inflationary models all energy densities apart from the inflaton are redshifted away and are thus negligible. This implies that there would be no plasma of photons and charged particles during the period of inflation.

The Cosmic Microwave Background (CMB) is still in evidence today and surprisingly we can see its remnants in something as mundane as tuning a television set where about one percent of the static observed on the screen is attributable to the CMB.

In 1965 a mysterious background noise on their radio antenna led Arno Penzias and Robert Wilson from AT&T Bell Laboratory to discover the CMB [51] which substantiated the Big Bang theory. The Cosmic Background Explorer (COBE) [52] satellite was launched in 1990 to measure the spectrum of the CMB radiation and confirmed the expected temperature fluctuation from place to place in the sky. The subsequent CMB experiments progressively improved on these initial measurements. This culminated in the launch of the Wilkinson Microwave Anisotropy Probe (WMAP) [53, 54] satellite in 2001 which has produced the most accurate measurement of the temperature fluctuations to date. This has lead to data that confirmed cosmological theory and can be used to determine the age, composition and geometry of the universe.

Peebles and Yu [55] successfully described the origins of the CMB as emerging from a universe which was a hot dense plasma composed of photons, electrons and protons. Figure 2.4 shows an overview of how the universe evolved over time. During the radiation-dominated phase, photons coupled to electrons through Thomson scattering while the electrons were coupled to the baryons through coulomb interaction. This formed a tightly coupled system known as the photon-baryon fluid. As the universe expanded so it cooled until at $T \sim 3000$K, known as "Recombination" ($z = 1089$), neutral Hydrogen was formed. At this point the photons decoupled and began free streaming towards a present day observer who would be able to note the small changes in temperature over the sky. It is these temperature differences on the surface of last scattering that account for the anisotropies we see[5] in the CMB today.

Inflation [56] predicts an exponential growth of the universe which smoothed out the majority of inhomogeneities. This is supported by the findings of COBE which found remarkable isotropy across the entire sky at one part in 10^5. It is believed that quantum fluctuations grew from a subatomic scale to an astrophysical size during inflation. Once inflation ceased these fluctuations became the density perturbations which gave rise to the observed anisotropies.

The tug-of-war between the attraction of gravitation and repulsion of radiation lead to an interplay where acoustic waves where produced in the coupled photon-baryon fluid. If we imagine the sound waves oscillating with time through the primordial plasma from inflation until recombination. The regions of longitudinal compression resulted in an increase in temperature while an alternating decrease in temperature occurred in the areas of rarefaction.

The size of these oscillations across the two-dimensional spherical surface can be quantified as an expansion in terms of spherical harmonics, Y_{lm}, of the temperature

[5] These photons are detected in the microwave band with a temperature of ~ 2.7K.

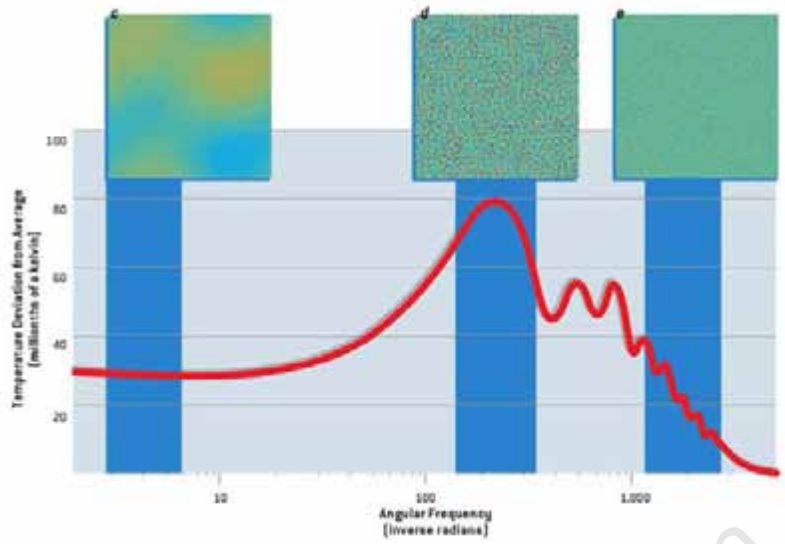


Figure 2.5: **Angular Power Spectrum of CMB [5]** – This shows that at large scales (c) the variations are not very noticeably but there is a strong peak depicting the largest contrast in variation at angels of about one degree across (d). This maximum then gradually decreases as we go to ever smaller scales (e).

variation, $\Delta T/T$, at a specified location,

$$\frac{\Delta T}{T} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_{lm} Y_{lm}(\theta, \varphi), \tag{2.23}$$

where the a_{lm} are the expansion coefficients and can be computed from the CMB map via:

$$a_{lm} \equiv \int Y^*_{lm}(\theta, \varphi) \frac{\Delta T}{T} d\Omega. \tag{2.24}$$

The *spherical harmonics* are defined as:

$$Y_{lm}(\theta, \varphi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P^m_l(\cos\theta) e^{im\varphi}, \tag{2.25}$$

where P^m_l are the associate Legendre functions with l and m the integers such that $l \geq 0$ and $|m| \leq l$. These spherical harmonics form a complete orthonormal basis i.e.

$$\int Y_{lm}(\theta, \varphi) Y^*_{l'm'}(\theta, \varphi) d\Omega = \delta_{ll'} \delta_{mm'}, \tag{2.26}$$

where $d\Omega = \sin\theta d\theta d\varphi$. Each multipole is given by the integer value of l where $l = 1$ is the monopole, $l = 2$ is the dipole etc. This is depicted in Figure 2.6. A higher multipole represents a smaller angular separation upon the sky, with $\theta \approx 180°/l$.

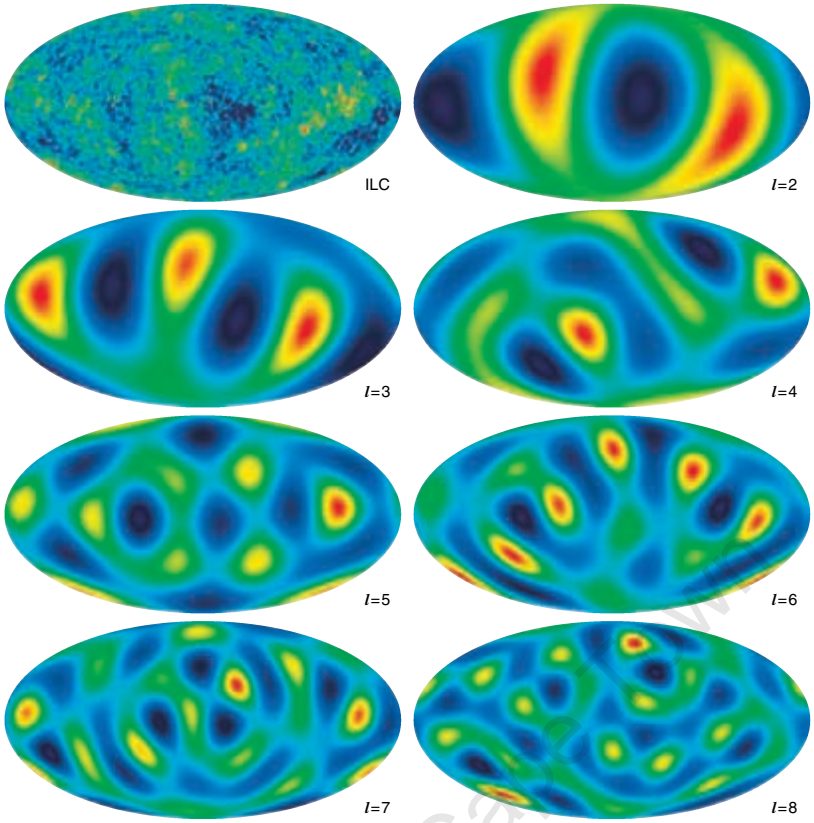


Figure 2.6: **Multipoles of the CMB** – Shown here (Credit[6]) are fits of the low order CMB multipoles from $l = 2$ to $l = 8$ for the WMAP data.

The C_l's can be defined in terms of the a_{lm}'s:

$$\langle a_{lm} a^*_{l'm'} \rangle = \delta_{ll'} \delta_{mm'} C_l, \tag{2.27}$$

where $\langle a_{lm} \rangle = 0$. This allows us to define $C_l = \sum_m |a_{lm}|^2$ and makes for a simple means to plot the power spectrum in terms of $l(l+1)C_l$ versus l. As shown in Figure 2.5 where the first predominant peak is at $l \sim 200$, which equates to about 1° across. The accompanying box figures overlayed inside the angular power spectrum plot show the temperature fluctuations for large scales (c), 1° - which shows the sharpest contrast in fluctuations - (d) and very small scales (e).

The photons that are streaming from the surface of last scattering have had little interaction with matter since decoupling, resulting in a clear snapshot of how the universe looked when it was 380,000 years old. There are however two weakly contributing effects that can alter the anisotropies observed. The Integrated Sachs-Wolfe (ISW) effect [57] results in the free streaming photons being redshifted (in overdense

[6] map.gsfc.nasa.gov.

regions) or blueshifted (in underdense regions) as they pass through gravitational potential wells on their path to an observer [58]. While the Sunyaev-Zel'dovich effect is due to inverse Compton scattering which transfers some of the energy from electrons to CMB photons, see [59] for a review.

Measurements made from the first peak of the angular power spectrum point to a universe that is spatially very close to flat. In addition the data from the CMB experiments can help constrain several cosmological parameters in particular when used in combination with other cosmological probes. Importantly any new cosmological model proposed that may account for an exotic form of dark energy or perhaps a particular parametrisation of dark energy must reproduce an angular power spectrum comparable to that measured by the most recent WMAP data [11]. Code has been developed such as CAMB [60] and CMBFAST [61,62] that can help in quickly testing the plausibility of new ideas by producing an angular power spectrum which can be compared to the results of WMAP. This technique is used in Section 4.5.1 where a double-step function parametrisation for dark energy in combination with other constraints is compared to the data for WMAP.

2.4 Supernovae

The two forces in conflict in the universe are the attractive pull of matter and the repulsive push of dark energy. The most dominant will be the deciding factor in whether the universe is slowing down or speeding up. It is not clear what the nature of dark energy is, its density is expected to change slowly or not at all as the universe expands. Although in the past the density of matter was greater than dark energy, at present the density of dark energy is in the ascendancy.

Type Ia Supernovae (SNe) are a well suited astronomical candidate to help establish the expansion history of the universe since they have an intrinsic luminosity. A Type Ia SN is believed to originate from a white dwarf star which accretes material from an orbiting companion until it reaches the Chandrasekhar limit. The result is a very bright explosion outshining the host galaxy and can easily be seen from ground based telescopes while space based telescopes are able to detect more distant events. The luminosity function is relatively well understood and calibrated making it possible to deduce the distance the supernovae occurred at. In contrast Type Ib, Ic, IIP and IIL SNe are not as reliable as Type Ia SNe at being standard candles due to the wider range of luminosities that they produce upon exploding, as opposed to the calibrated and narrow luminosity range emitted for Type Ia supernova.

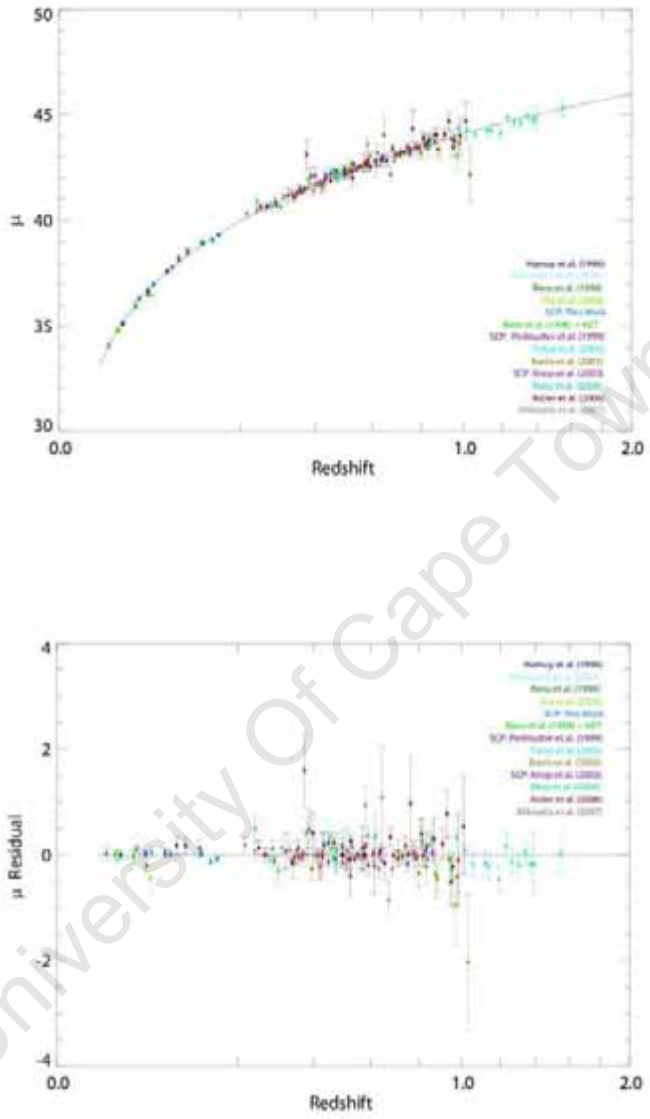


Figure 2.7: **Hubble Diagram from Union Supernova Data [6]** – In the top panel the distance modulus is shown of the Union Sample data from Type Ia SNe constructed with selection cuts and outlier rejection. Each sample is independently binned in redshift bins of 0.01 with the line being a best fit for $\Omega_m = 0.29$ and $\Omega_{\rm DE} = 0.71$. The bottom panel shows the residual of the distance modulus.

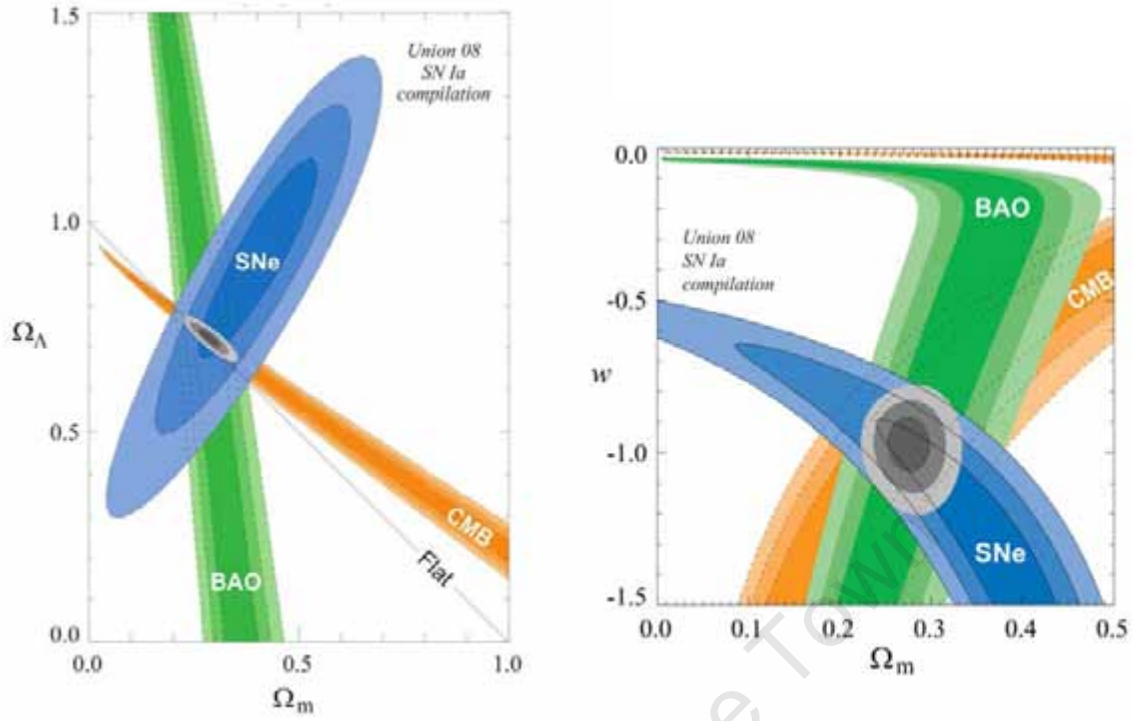


Figure 2.8: **Combined Cosmological Probes [6]** – The constraints for Ω_Λ and Ω_m are shown in the left panel while w and Ω_m are shown in the right panel. Confidence levels of 68%, 95%, and 99.7% are depicted in progressively lighter shades. The error ellipses are obtained from CMB (orange), Baryon Acoustic Oscillations (BAO) (green), and the Union Compilation of SNe Ia data (blue). The intersecting region produces a very tightly constrained error ellipse from the combined data (grey).

Riess et al. [63] and Perlmutter et al. [64] concluded from measurements of Type Ia SNe that the universe is expanding at an accelerated rate. This was a profound discovery that lead to the resurgence of the cosmological constant, see Section 2.5.1. High redshifts SNe are required to substantiate that the current burst of acceleration is only a recent phenomenon. The chance discovery of an ancient (high redshift) Type Ia supernova by Riess 2001 [65] on the Hubble Space telescope helped confirm that there was an initial slowdown of the universe's expansion before entering the current late period of acceleration. Figure 2.7 shows a recent compilation of Type Ia SNe data in a distance modulus diagram. These results can also be used in combination with other cosmological probes to help contrain cosmological parameters and dark energy, as shown in Figure 2.8.

The Dark Energy Task Force (DETF) [66] outlines a program for current and

future surveys in Stages[7] which will help understand and better constrain dark energy. Amongst the DETF proposed supernova surveys for Stage-IV are the Supernova Acceleration Probe (SNAP), Joint Dark Energy Mission (JDEM) and Large Synoptic Survey Telescope (LSST). These will provide improved accuracy in measuring the distance modulus and are used in Section 4.3.1 to see if dynamical dark energy can be detected at low redshift.

2.5 Theoretical Models of Acceleration

In Chapter 4 we will study scalar field models of late time acceleration. It is therefore appropriate at this point to give a basic introduction to theoretical models of acceleration including the cosmological constant and scalar fields in cosmology.

2.5.1 Cosmological Constant - Λ

As discussed in Section 2.1 the Einstein equations reduce to the Friedmann equations. One of Einstein's primary interest was to introduce a static universe with positive energy density. This is possible if you consider a positive curvature, $+k$, in Eq. (2.8) but does not resolve the problem in Eq. (2.9) where $\ddot{a}$ remains non-vanishing since the pressure p is also considered non-negative. A simple fix was to introduce a modification to Einstein's equations to create a static universe:

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}\mathcal{R} + g_{\mu\nu}\Lambda = 8\pi G T_{\mu\nu}, \tag{2.28}$$

where the infamous Λ is introduced as the new parameter. Considering the Friedmann equation with this modification produces,

$$H^2 = \frac{8\pi G}{3}\rho + \frac{\Lambda}{3} - \frac{k}{a^2}, \tag{2.29}$$

while the acceleration equation becomes,

$$\frac{\ddot{a}}{a} = -\frac{4\pi}{3}(\rho + 3p) + \frac{\Lambda}{3}. \tag{2.30}$$

The discovery of the expanding universe by Hubble did away with the original motivation for constructing a static universe and the cosmological constant was renounced by Einstein. There are two key shortfalls that still haunt the idea of the

[7]Stage-I represents what is known from past surveys. Stage-II is the status after completing current surveys. Stage-III surveys comprises of near-term, medium-cost, current proposed projects while Stage-IV are larger long term projects.

cosmological constant. The first is that the terms in Eq. (2.30) are so precariously balanced that it would only take a very slight deviation to lead to a catastrophic runaway departure from a static universe.

The second and more recent problem relates to the fact that the cosmological constant should be equivalent to the energy density of the vacuum. The measured energy density is however drastically smaller than any theoretical prediction, the scale of the difference has been described as the most embarrassing problem in physics.

2.5.2 Scalar Field Theory

Scalar field theory has become an important component of modern Cosmology with well known applications to modeling the matter content of the early inflationary universe with a non-negative self interacting potential $V(\phi)$. A detailed review of scalar fields is given in [20, 42, 67] while only a cursory overview will be given here to familiarise the reader with the required framework used in Chapter 4.

The life of a scalar field begins by introducing the action over a spacetime,

$$S = -\int \sqrt{-\det g_{\mu\nu}}\,\mathcal{L}\; d^4x, \tag{2.31}$$

where the Lagrangian of a standard, minimally coupled, scalar field is given by:

$$\mathcal{L} = \frac{1}{2}g^{\mu\nu}\partial_\mu\phi\partial_\nu\phi - V(\phi), \tag{2.32}$$

and the potential is denoted by $V(\phi)$. The energy-momentum tensor for a scalar field is found by varying metric [42]:

$$T^{\mu\nu} = -g^{\mu\nu}\left[\frac{1}{2}g^{\rho\sigma}\partial_\rho\phi\partial_\sigma\phi + V(\phi)\right] + g^{\mu\rho}g^{\nu\sigma}\partial_\rho\phi\partial_\sigma\phi. \tag{2.33}$$

Given you are considering a FLRW metric, these results match the energy-momentum tensor for a perfect fluid with density and pressure given by,

$$\rho_\phi = \frac{1}{2}\dot{\phi}^2 + V(\phi), \tag{2.34}$$

$$p_\phi = \frac{1}{2}\dot{\phi}^2 - V(\phi). \tag{2.35}$$

Using the above relationships for p and ρ and substituting them into Eq. 2.10, which ensures energy-momentum conservation, we can derive the evolution equation for a

homogeneous scalar field in an expanding universe,

$$\ddot{\phi} + 3H\dot{\phi} + \frac{dV}{d\phi} = 0. \tag{2.36}$$

As previously discussed the ratio of the pressure to energy density is defined as the equation of state parameter for the field and is given by $w_\phi \equiv p_\phi/\rho_\phi$.

The strong energy condition translates into a constraint on the equation of state parameter:

$$w_\phi > -\frac{1}{3}, \tag{2.37}$$

but to ensure that a scalar field results in an accelerating universe the strong energy condition must however fail, i.e. $w < -1/3$. The choice of the scalar field's potential determines the dynamics of the field. We next introduce a particular scalar field which generically arises in particle physics theories involving compactified dimensions [67].

Scaling Field - Tracking Model

A scaling model (also called a tracking model), first introduced by [68], is a scalar field with a potential that allows it to mimic the dominant energy component of the universe. An example is the exponential potential [69], which we discuss in detail below. A tracking model results in a fixed fractional contribution that Ω_ϕ makes to the total energy density for each appropriate epoch. Figure 2.9 shows, in the left panel, the nature of a tracking model with the equation of state parameter matching the dominant phases in the universe's evolution. In the right panel of Figure 2.9 we can see how the energy density contribution of $\Omega_{\rm DE} = \epsilon$ remains constant until at z_t it begins to grow at the onset of a period of dark energy dominance.

A tracking model arises when we consider a canonical scalar field, ϕ, minimally coupled to gravity, with an exponential potential,

$$V(\phi) = M^4 e^{-\lambda\kappa\phi}, \tag{2.38}$$

as given by [70] where M is a constant with dimensions of mass, λ is a dimensionless constant, $\kappa^2 = 8\pi G = 8\pi m_{\rm pl}^{-2}$ and $m_{\rm pl} = 1.22 \times 10^{19}$ GeV is the Planck mass.

Consider a field evolving in a spatially flat FLRW universe containing a fluid with a barotropic equation of state $w = p/\rho$, where p and ρ are the respective pressure and energy density of the fluid. This results in two attractor solutions for the scalar field, which depend on the values of λ and w_b [70]. In the case where $\lambda^2 > 3(1+w)$ the scalar field will enter into a scaling regime, following the evolution of the background fluid with $w_\phi = p_\phi/\rho_\phi = w$ and $\Omega_\phi = 3(1+w_b)/\lambda^2$. If, however, $\lambda^2 < 3(1+w_b)$ then

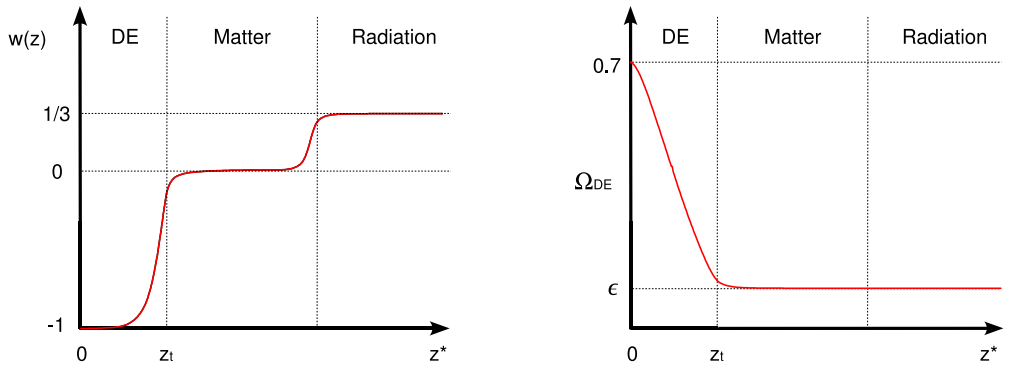


Figure 2.9: **Tracking Scalar Model** – In the left panel is shown how the dark energy equation of state parameter $w(z)$ tracks the dominant epochs of radiation and matter from recombination z^* until z_t. In the right panel is a schematic illustration of how the corresponding energy density of dark energy, $\Omega_{\rm DE} = \epsilon$, remains the same from radiation through to the matter dominated eras until it begins to grow at a redshift z_t where it stops tracking and enters the dark energy dominated phase and continues to increase until it reaches a value of $\Omega_{\rm DE}(z = 0) = 0.7$, today. In order to achieve this behaviour $w(z)$ must be negative for a sufficiently long period of time.

the field will be the dominant component of the universe, with $w_\phi = -1 + \lambda^2/3$ and $\Omega_\phi = 1$.

Consider an early time constraint that can be placed on the value of the energy density of the field, Ω_ϕ, taken in combination with a scaling field model. This will set the value of Ω_ϕ to be constant throughout the radiation and matter dominated epochs until, at the dark energy phase of dominance, it starts growing to reach the energy density of ~ 0.7 measured today, illustrated in the right panel of Figure 2.9. In the next section we discuss Big Bang Nucleosynthesis which can give rise to such early time constraints [71].

One known shortfall of the single exponential potential scalar field is that it doesn't allow for a late phase of acceleration. This can be addressed in two possible ways. The first is to employ a particular parametrisation for the equation of state from the point that dark energy becomes dominant. The second is to introduce a double exponential scalar field potential, as described by [72], which still produces tracking but does allow for a period of dark energy dominance to arise naturally from the scaling field. Both these methods are further discussed and used in Chapter 4.

2.6 Big Bang Nucleosynthesis

The theory of Big Bang Nucleosynthesis (BBN) is concerned with the origin and abundance of light and heavy nuclei. Gamow, Alpher and Herman [73] proposed the hot Big Bang to explain the existence of all elements while Burbidge, Burbidge, Fowler and Hoyle [74] calculated the nucleosynthesis process which takes place in stars and generates elements heavier than Helium over a long time scale. Today we understand that both these mechanisms contribute to the formation of elements. The majority of the Helium is forged in the Big Bang while carbon and everything heavier is the product of nucleosynthesis in stars.

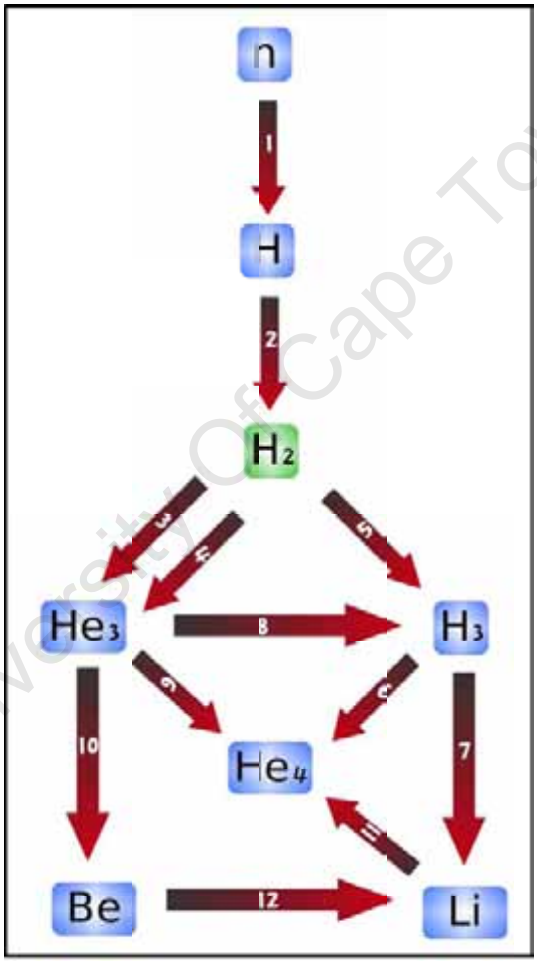


Figure 2.10: **BBN Reactions** – Here the hierarchy of reactions leading to the production of the light nuclei is shown for Big Bang Nucleosynthesis. The numbers correspond to the reactions listed in Table 2.2. Central to the production of more complex elements (from 3 onwards) is sufficient production of deuterium, H^2 (shown in green), being available.

The sequence of elements formed during BBN are illustrated in Figure 2.10 with

the corresponding numbered reactions listed in Table 2.2. The fractional abundance of the different elements are depicted in Figure 2.11 as a function of time (top axis) and temperature (bottom axis).

1	$n \longrightarrow H^1 + e^- + \overline{\nu}$	7	$H^3 + He^4 \longrightarrow Li^7 + \gamma$
2	$H^1 + n \longrightarrow H^2 + \gamma$	8	$He^3 + n \longrightarrow H^3 + H^1$
3	$H^2 + H^2 \longrightarrow He^3 + n$	9	$He^3 + H^2 \longrightarrow He^4 + H^1$
4	$H^2 + H^2 \longrightarrow H^3 + H^1$	10	$He^3 + He^4 \longrightarrow Be^7 + \gamma$
5	$H^2 + H^3 \longrightarrow He^4 + n$	11	$Li^7 + H^1 \longrightarrow He^4 + He^4$
6	$H^2 + H^3 \longrightarrow He^4 + n$	12	$Be^7 + n \longrightarrow Li^7 + H^1$

Table 2.2: **Order of Reactions for BBN** – Here the chemical reactions undergone during BBN are listed roughly in the order that they occur. The numbers correspond to those listed in Figure 2.10.

Directly after the Big Bang and baryogenesis the universe was hot, fully ionised and dissociated. This resulted in a thermal equilibrium between protons and neutrons. As the universe expanded the temperature cooled until, at $T \sim 0.8$ Mev [75], the expansion rate matched and subsequently exceeded the weak interaction rate of protons and neutrons. This meant that the equilibrium could not be maintained any longer and a fixed ratio of neutrons to protons, 1:6, emerged at a stage called freeze out. No further reactions were possible at such a high temperature allowing the neutrons an opportunity to decay into protons. This pushed the final ratio of neutrons to protons up to 1:7.

Approximately 100 seconds after the Big Bang the temperature had sufficiently decreased to allow for deuterium to start forming. This is a fundamental building block for further reactions to produce heavier elements, as can be seen from Figure 2.10. The amount of heavier elements that can be produced are thus dependent on the amount of deuterium that is available. Deuterium is however limited by the ratio of neutrons to protons (set by the expansion rate) and is only produced for a certain period of time while the temperature is sufficiently high.

Initially any deuterium formed is broken up again since the background energy is higher than the binding energy of deuterium. This is known as the deuterium bottleneck since it limits the production of deuterium until the temperature cools sufficiently to allow for stable deuterium to remain. A portion of deuterium produced is then swept up into the production of He^3 and H^3 until the continual cooling of the

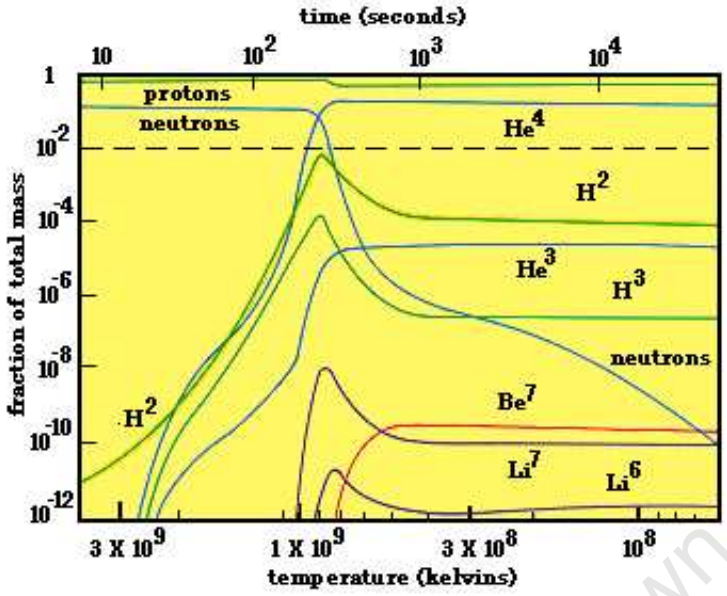


Figure 2.11: **Evolution of Fractional Composition of Elements during BBN** – Here the evolution of the fractional composition for nuclei abundance (left axis) is shown shortly after the Big Bang, as a function of time (top axis) and the associated temperature of the universe (bottom axis). Helium is produced after only approximately 20 seconds (∼1 MeV) which is followed by smaller amounts of other light elements (Beryllium, Lithium). The results in this figure (Credit[8]) are based on calculations from [7].

universe leads to a temperature that is too low to allow these reactions to continue. Figure 2.11 illustrates this sequence of events showing the deuterium beginning to peak after 100s along with the subsequent production of heavier elements.

This represents a snapshot of the particle and element production in a given period of time (with a corresponding expansion rate) and the abundances are then set for the rest of the universe's existence. The exceptions are Be^7 and H^3 which undergo radioactive decay to form He^3 and Li^7 with a half-life of 12 years and 53 days respectively and hence traces of these elements don't exist from primordial origin today. The majority of the protons that did not end up forming part of the more stable He^4 remained as free particles. This resulted in an abundance of 25% Helium and 75% Hydrogen by mass as measured today. Heavier nuclei form only trace components of the total abundance.

The ratio of protons to neutrons is the key indicator of what the final abundance of He^4 is after BBN and is thus exponentially sensitive to the expansion rate of the universe. Thus tampering with any of the cosmological parameters that effect this

[8] www.astro.ucla.edu/~wright/cosmology.htm.

ratio (expansion rate) can, lead to results that don't match the measured abundance. This allows a very accurate means to pin down the expansion rate of the universe.

If, for example, there is a large concentration of dark energy at $T \sim 1$ MeV the expansion rate of the universe would be significantly larger which would lead to an earlier freeze out altering the neutron to proton ratio and thus changing the final fraction of He^4 produced. Bean et al. [71] show that this allows a constraint of $\Omega_{\rm DE}(z_{\rm BBN}) < \epsilon = 0.045$. This result together with the tracking model of Ferreira et al. [69] and the analysis of Doran et al. [76] provide the main starting point for our analysis in Chapter 4, where we study the implications of this BBN constraint on dark energy dynamics at low redshifts.

2.7 Conclusion

This chapter presented a self contained summary of cosmology with specific topics that lay the foundations for the work presented in Chapter 4 and 5. In the first of these chapters we consider the constraints that BBN imposes on late time dark energy dynamics when a tracking model is assumed. While in Chapter 5 we highlight the novel use of a new software package called Fisher4Cast which can produce unique visualisations of Fisher Matrix techniques in cosmology.

Chapter 3

Optimization

Optimization is an area of great diversity with broad applicability to industry and research. Other examples where optimisation is becoming more and more important in cosmology and astronomy are related to automated searches of large datasets, see e.g. [77, 78]. Searching for unusual or weird objects is an optimisation problem and improving the efficiency of the underlying optimisation algorithm is an important way of maximising the science from future surveys.

It is common in many paradigms to use a very small but well understood group of standard optimization algorithms. These are simple and proven methods of solving complex non-linear optimization problems. In this chapter we review the idea of combining some of the standard algorithms to form a hybrid style algorithm [26]. This is also examined in combination with a multiple-try sampling technique [79] which results in a completely new algorithm, Hybrid-MTM, that leads to impressive gains and shows particular applicability to higher dimensional problems.

In the introductory chapter we covered the concept of optimization and introduced some of the well known and commonly used heuristic algorithms. Next we propose three commonly used standard test functions used in the context of evaluating optimization algorithms. We then compare both the efficiency and accuracy of these methods with a hybrid algorithm which incorporates aspects of Particle Swarm Optimization (PSO) and Simulated Annealing (SA). This is done by evaluating each algorithms performance on the set of test problems introduced over a range of dimensions (5 to 50 dimensions) where we look for the best convergence rate to the minimum after a set number of steps for each of the algorithms. We also look at the Central Processing Unit (CPU) time used in each instance. A vast improvement is

observed and shows that there exists a great deal of promise for a Hybrid algorithm.

The addition of a new sampling method, called Multiple-Try Metropolis (MTM), is also considered. This is first used in conjunction with the well known Monte Carlo Markov Chain (MCMC) algorithm and then applied to the newly proposed Hybrid method. This produces significant improvements with particular applicability to higher dimensional problem where a billion fold improvement is gained in certain instances over the commonly used methods.

3.1 Benchmark Problems

There exists a set of well known test problems which are commonly used to apply optimization algorithms to. These provide good benchmarks for both testing the efficiency of new algorithms and comparing a range of optimization algorithms against each other.

3.1.1 Rosenbrock

The Rosenbrock function has a distinct parabolic shaped valley shown in Figure 3.1. To find this valley is relatively easy but to converge on the actual global minimum contained within the valley is far more challenging. The Rosenbrock functions is given by,

$$R(x) = \sum_{i=1}^{N-1} \left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2\right] \quad \forall x \in \mathbb{R}^N, \tag{3.1}$$

where N is the number of dimensions being considered. The global minimum of this function is given at $(x_1, x_2, \ldots, x_N) = (1, 1, \ldots, 1)$ in general but there also exists a local minimum for dimensions $N > 4$ at $(x_1, x_2, \ldots, x_N) = (-1, 1, \ldots, 1)$.

3.1.2 Paraboloid

The Paraboloid function, shown in Figure 3.2, is a simple example and can easily be solved by a deterministic Conjugate gradient type of algorithm but yet offers sufficient complexity to test a heuristic algorithm. This becomes particularly relevant when higher dimensional instances are considered. The equation for the Paraboloid is given by,

$$H(x) = \sum_{i=1}^{N} x_i^2, \tag{3.2}$$

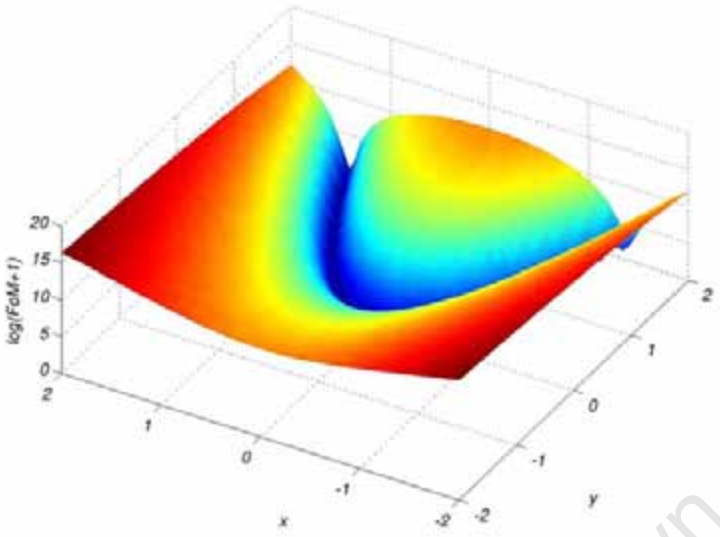


Figure 3.1: **Rosenbrock function** – Shown here is a 2D plot of the Rosenbrock function where the z-axis gives the logarithm of the Figure of Merit (FoM), log(FoM+1), in order to highlight the parabolic valley in which the global minima is hidden.

where N is the number of dimensions. The global minima is given by $(x_1, x_2, \ldots, x_N) = (0, 0, \ldots, 0)$.

3.1.3 Griewangk Function

The Griewangk function has an overall shape very similar to the Paraboloid test function but in addition has small and deep dimples which are spaced evenly across the entire hypersurface. There exists two scales to this problem, one is a long range curvature to the function which is centered around the global minimum and the second is a short range curvature which is given by the dimples or local minima. In most instances these two scales are sufficiently contradictory to render a heuristic algorithm very ineffective. The equation for the Griewangk function is given by,

$$G(x) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \tag{3.3}$$

where N is the total number of dimensions. The global minimum exists at $(x_1, x_2, \ldots, x_N) = (0, 0, \ldots, 0)$ while there are multiple local minima.

The methodology employed when applying an optimizing algorithm to one of these test problems is to start from a unique position for each function and then let the chain evolve from there. The progress of the algorithm is summarised by plotting the

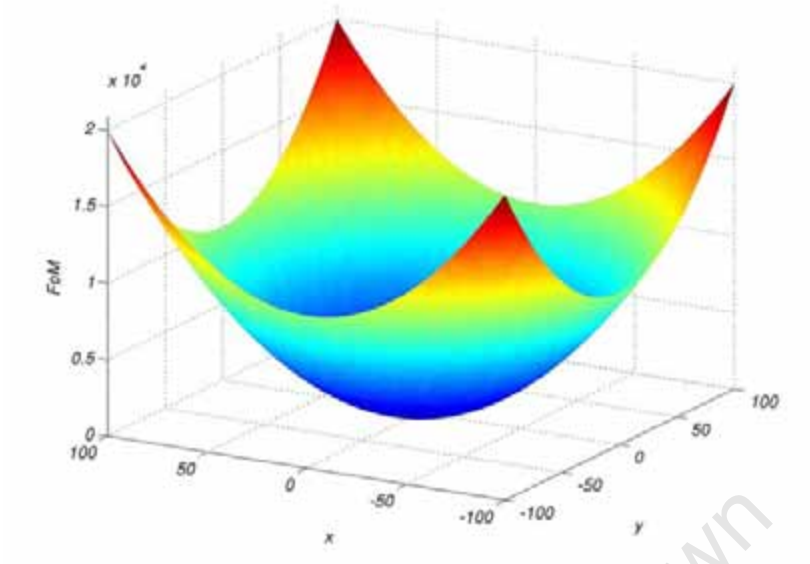


Figure 3.2: **Paraboloid function** – A 2D Paraboloid with the z-axis depicting the FoM and showing the huge scale that is required to cover before converging to the minima at the center of the function.

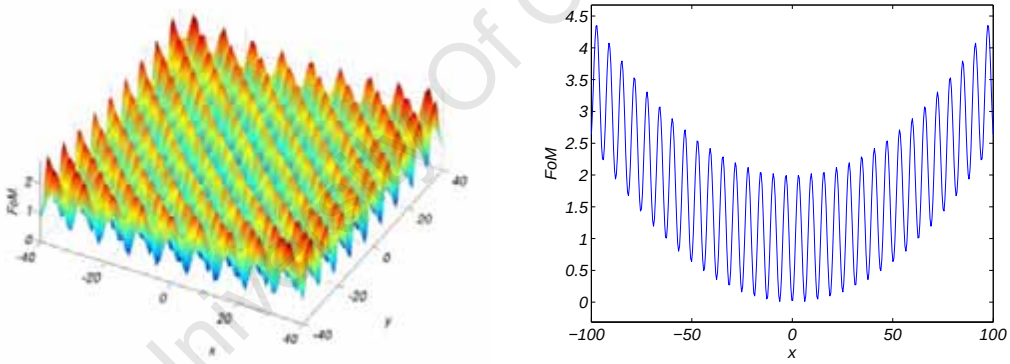


Figure 3.3: **Griewangk function** – A 2D plot of the Griewangk hypersurface where the z-axis gives the Figure of Merit (FoM), showing the small scale minima or dimples in the left panel while the overall larger scale curvature is shown in the 1D slice through the x-axis in the plot on the right.

FoM at each step for a fixed number of steps. In order to get a true reflection of the performance a total of 50 runs were done for each of the algorithms on each of the test problems. In each instance the best FoM was recorded at every step and used for the plots comparing the FoM. This does away with the need for error bars since we are only plotting the best results for each test case and are not showing an average.

3.2 Introduction to Hybrid

A unique and promising algorithm which combines aspects of both Particle Swarm Optimization (PSO) and Simulated Annealing (SA) is aptly named Hybrid [26]. As opposed to one chain, Hybrid has multiple chains running at the same time which share information about their position and what the best FoM is amongst the chains much the same as PSO. This enables each chain to evaluate how its relative FoM, at a certain position, relates with the other chains and then use this information as a basis to determine the nature of both the range for a new step and the transition probability of accepting that new position.

Hybrid incorporates the idea of a traditional cooling schedule, α, applied to the transition probability as used in SA and in addition applies this idea to the step size as well. This is implemented by using a multiplicative factor which governs the step size of all the chains. It gradually becomes smaller in size which in turn tempers and reduces the step size of all the chains as the number of iterations increase.

To summarise the rational of the Hybrid algorithm. The chains share information about their relative position and FoM in a collaborative sense. If a particular chain's FoM is good relative to the other chains then it will start to take smaller steps since it would assume that it is close to the optimum position and needs to refine its search. In contrast if a particular chain is doing very badly relative to the rest of the chains, then it would start taking more bold, large steps to allow it the opportunity to more broadly search the space for a favourable position.

Hybrid uses two functions, f and g, to dynamically adapt the proposed step size of each chain. This is achieved by multiplying both functions with σ_i^0, which is drawn from a Gaussian distribution centered around the current position,

$$\sigma_i(j) = \sigma_i^0(j) \times f_i(j) \times g(j). \tag{3.4}$$

Here we have employed the same notation as introduced for PSO in section 1.2.1 where the subscript i represents the chain and j the step under consideration. We first define a ratio of the FoM for a particular chain, i, relative to the average FoM of all the other chains,

$$p_i(j) \equiv \frac{\mathrm{FoM}_i(j)}{\sum_{k\neq i}^{N} \mathrm{FoM}_k(j)/N}, \tag{3.5}$$

where N is the total number of chains and we have used a shorthand notation for the FoM of a given chain, i, at a given position x after j steps, $\mathrm{FoM}_i(j) \equiv \mathrm{FoM}(x_i(j))$. This acts as an indicator of how well a single chain is doing compared to the rest

of the chains. If a chains current position resulted in a bad FoM compared with the rest of the chains the ratio would be $p > 1$ (in the case of minimization) while if it were doing relatively well $p < 1$. The piecewise function f is thus chosen to use the information given by the ratio of p,

$$f(j) = \begin{cases} p_i(j)^{-\gamma} & \text{where } \ 0 < p \leq 1, \\ \\ mp_i(j) + c & \text{where } \ 1 < p \end{cases} \tag{3.6}$$

where $f(j) = f(p_i(j))$. A cutoff is employed for p where the ratio is only considered up to a maximum of 2. This limits the maximum contribution of f to Eq. (3.4) and thus the largest likely step size to be taken for a chain that has a very bad relative FoM. This prevent the instance of a runaway bad chain which in turn has the negative effect of "freezing" the progress of the other chains since their ratio p becomes very small, since the average FoM of the chains becomes unavoidably large. This producing a small f which in turn results a small σ from Eq. (3.4), producing a set of chains which make very small step proposals. Eq. (3.6) is a refinement of the equation for $f(j)$ first introduced in [26].

The factor q acts as a means to gauge the current average FoM, $\langle \text{FoM}(j) \rangle$, relative to the initial average FoM, $\langle \text{FoM}(1) \rangle$,

$$q \equiv \frac{\langle \text{FoM}(j) \rangle}{\langle \text{FoM}(1) \rangle}, \tag{3.7}$$

where $\langle \text{FoM}(j) \rangle \equiv \sum_{k=1}^{N} \text{FoM}_k(j)/N$. The factor q can then loosely be thought of as the memory of the chains which give an indicator of how well the chains as a whole are doing at improving the average FoMs of the chains relative to their initial average FoM from their starting positions. The factor g is given by,

$$g = q^{\beta}, \tag{3.8}$$

which gradually tempers and refines the step size to become smaller as q improves (becomes smaller) or help increase the step size in the event that q becomes larger. Both f and g can be tailored to suite the nature of the problem under consideration. Tables 3.1, 3.2, 3.3 list the parameters for m, c, γ and β used in Eq. (3.6) and Eq. (3.8) for each of the benchmark problems to which Hybrid was applied.

The transition probability is much the same as introduced in, Section 1.2.1, on

SA:

$$P_i(x(j+1)|x(j)) = \exp\left[-\alpha(x_i(j+1) - x_i(j))\right], \tag{3.9}$$

where α is inversely proportional to the temperature for SA and set to $\log(1+j)/T_0$ as a standard cooling schedule.

If we look at SA and MCMC it turns out that these two algorithms are trivial to convert and run in a parallel implementation. Hybrid is not as straight forward unfortunately. The reason for this lies in the fact that it shares information between the chains while the parallel chains for SA and MCMC can be run completely independently. There may be a more adept version or implementation of Hybrid that would be best suited to parallelization but this is left as a question to be addressed in future work.

We next give a blueprint for writing a Hybrid algorithm by detailing step-by-step procedures in the pseudo code and accompanying flowchart, shown in Figure 3.4.

3.2.1 Pseudo code

The pseudo code for Hybrid follows a very similar structure to that of SA but with only two fundamental differences. The first being that there are multiple chains and the second being that additional variables are calculated and used to reposition a chain from one step to the next.

- Initialize: We begin by initializing the position and calculate the appropriate values for f and g given by Eq (3.6) and Eq (3.8). We also set the condition for convergence.
- Set α: Next we set the cooling schedule, $\alpha = \log(1+j)/T_0$, the same as used in SA.
- Propose move: Here we use Eq. (3.4) to calculate a newly proposed position.
- Is the proposed FoM better?:

  - If the newly proposed position has a lower FoM than the previous position, then accept the move to the new position. In the case of the test problems introduced in Section 3.1 the FoM is simply the function value returned for the current position of the chain for the benchmark problem it is being applied to.
- Randomly accept bad move?:

  - If the newly proposed position has a higher FoM than the current position a move wouldn't normally be allowed. In exactly the same way as we used the

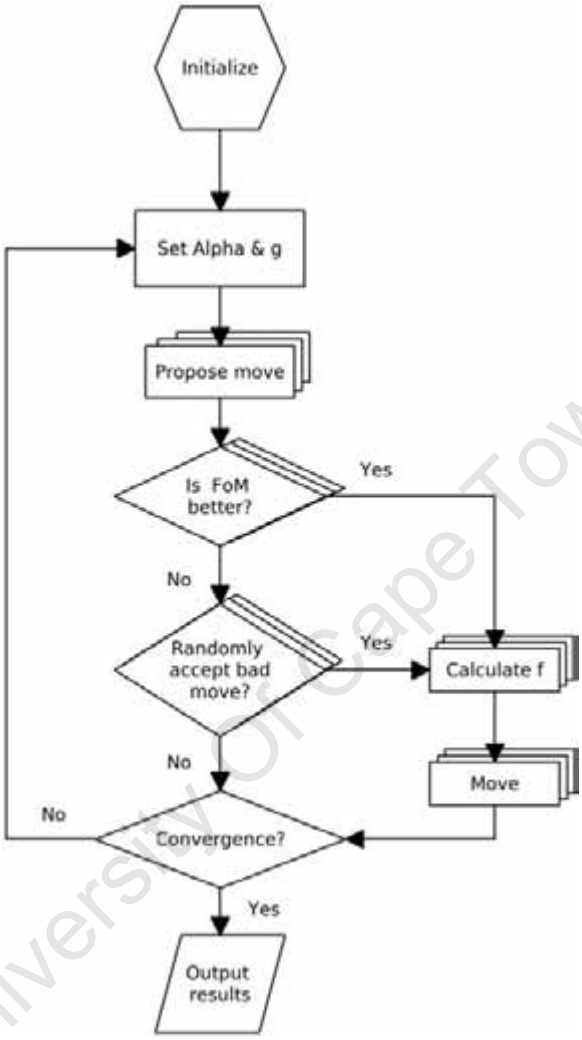


Figure 3.4: **Flowchart of Hybrid Algorithm** – This flowchart outlines the required components and sequence to build a Hybrid algorithm. The accompanying pseudo code in the text describes each element listed in the flowchart. The multiple layered functions in the flowchart indicate those specific subcomponent of the algorithm that are applied to each chain, i, individually. It owes its origin to the PSO and SA algorithms and the similarity with these flowcharts can be seen in Figure 1.9 & 1.2.1 shown in the introduction.

Metropolis Hastings algorithm to allow for bad moves in SA, we do the same for Hybrid. This helps to ensure that the system doesn't converge in local minima and the algorithm can "climb" its way out of bad positions to better explore the entire space for a global minimum. The probability is given by Eq. (3.9) and the move is thus accepted if a uniform random number is generated which is lower than the transition probability.

- Record data: This is an optional step not shown in the flowchart of Figure 3.4 but can be used to record data that can be used for postprocessing analysis. Data that can be recorded includes the lowest FoM found at each step and its position in the parameter space along with the α value at each iteration. The number of accepted and rejected steps can also be recorded to analyse what the acceptance ratio of proposed steps was. Other data that can be recorded include the factors for σ_i, p_i, f_i, q and g given by Eq. (3.4)-(3.8).

- Convergence?: Here we consider the convergence criteria which needs to be met before the while loop ends. There are a range of criteria that can be used for convergence. The simplest is to set a fixed number of iterations or a fixed length of time that the code runs for. More sophisticated methods would set a threshold of accuracy for the FoM or cease when the gradient of the FoM doesn't change significantly over a set number of iterations. If the while loop is not ended then the algorithm loops back to the second step. For the remainder of this chapter we use the simple method of limiting the number of steps (usually 1000 steps).

- Output results: This is done once the loop has been completed and the best FoM and its position along with the remaining statistics of the run can be output to the screen or a file.

3.2.2 Hybrid Applied to Test Problems

We will be using the Rosenbrock, Paraboloid and Griewangk benchmark problems introduced in Section 3.1. These will be used as a means to assess each of the optimization algorithms and thus be able to compare their relative performance. In order to make the problems particularly challenging we also vary the range of dimensions to include 5D, 20D, 35D and 50D cases. This will really test the algorithms and also minimize the possibility that an algorithm is tuned to a specific dimensional problem. All the results plotted in this chapter represent the lowest FoM at each step.

So if there are multiple chains and multiple runs, the chain with the lowest FoM at a given step of all the runs, is recorded as the minimum FoM and used in the plots. This also explains the lack of error bars for all the results plotted since they represent a minimum value after 50 runs and not an average.

The discussion in [26] showed both error bars and minima are correlated and that Hybrid gives less dispersion than the other methods because of the sharing of information. We decided to focus on the best of the 50 runs as a measure of performance. Tables 3.1-3.3 list the parameter values for each of the algorithms used for each of the respective benchmark problems. These parameters were chosen a-priori to consistently produce good results for any given test function and dimension without requiring a drastic change in the parameter values. An exact and exhaustive analysis of the best parameter values for given test problems is left for future work.

In Figure 3.5 a MCMC, SA and Hybrid optimization algorithm are run on the Rosenbrock test problem. Both the MCMC and SA algorithms are similarly matched and converge at approximately the same rate towards the minimum. The Hybrid algorithm is on par with these two algorithms in the case of the five dimensional example and shows some improvement in the higher dimensional examples. In the 50 dimensional case, this becomes more significant in particular the initial rate of convergence. The Hybrid algorithm is fundamentally stochastic. Sometimes a random jump leads to a significant improvement, while at other times it is possible for all chains to take worse steps.

	α	g^{β}	f: If $p > 1$	f: If $p < 1$	Chains	No. Runs
MCMC	0.5	-	-	-	1	20
SA	0.5	-	-	-	1	20
Hybrid	0.5	$g^{0.2}$	$13p + 0.2$	$p^{1.2}$	10	20

Table 3.1: **Rosenbrock Function** – This table lists the parameters used for MCMC, SA and Hybrid to produce the results plotted in Figure 3.5. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (-2, \ldots, -2)$ for all chains while the range was restricted to $x_i = [-2.048, 2.048]$.

For the Paraboloid problem we have the comparisons of the algorithms displayed in Figure 3.6. This again shows very similar performance between the MCMC and SA algorithms for all the dimensional cases considered. In the instance of the Hybrid algorithm there is a more marked improvement for the five dimensional case over the two other optimization techniques, however there is a drastic improvement for the

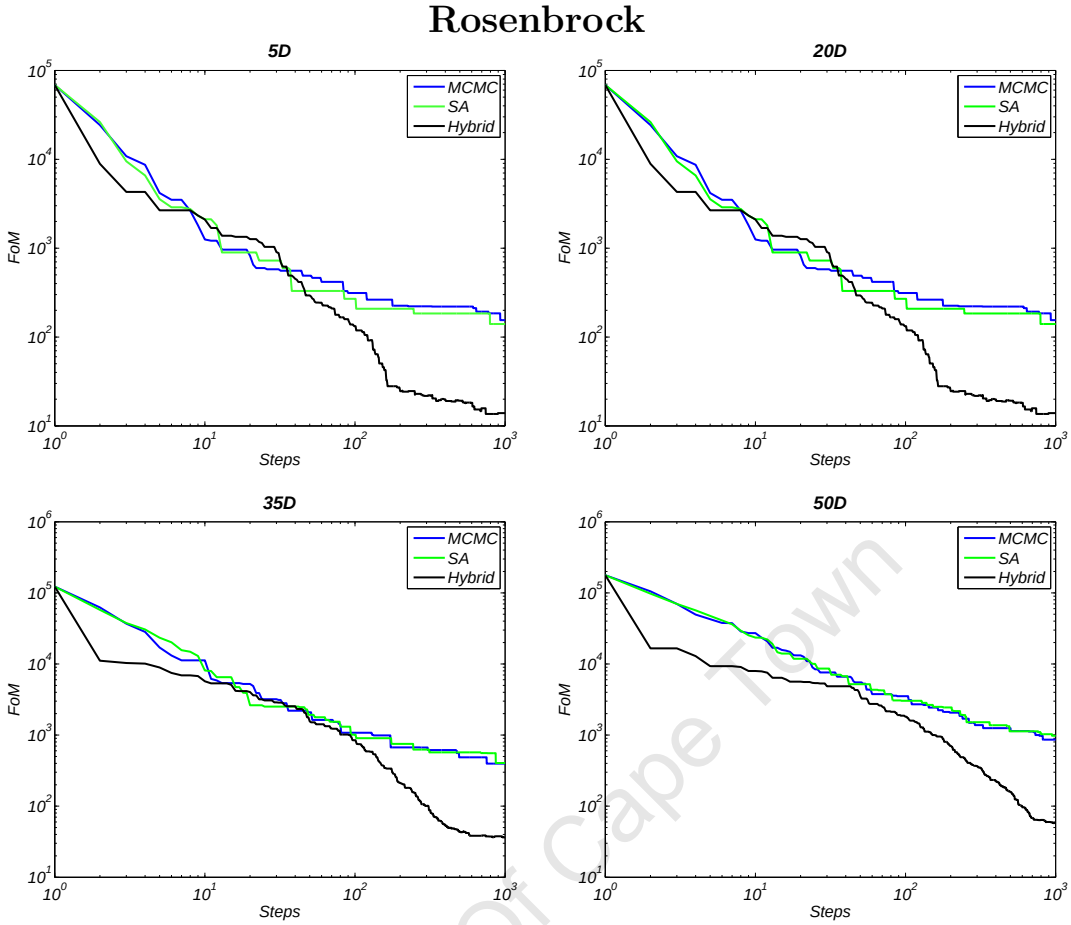


Figure 3.5: **Hybrid** - In the case of the Rosenbrock function Hybrid outperforms both SA and MCMC consistently, after 1000 steps, for all the dimensions considered. The FoM is the Rosenbrock function value and the lowest single FoM is plotted at each step amongst all the chains from the entire set of 20 runs generated.

higher dimensional cases. In the 20, 35 and 50 dimensional problem the other two algorithms struggle to converging to the minimum while Hybrid successfully manages to do this in each of these cases. The overall FoM is up to three orders of magnitudes better in the case of the 50 dimensional problem for Hybrid.

The comparative results for the more complex Griewangk test function are shown in Figure 3.7. Here the SA algorithm begins to marginally outperform the MCMC algorithm but this is vastly over shadowed by the impressive way in which the Hybrid heuristic completely outperforms the other two algorithms. As we have seen in the two other test problems Hybrid performs progressively better as we consider higher dimensional instances of the test function. The MCMC and SA algorithm both struggle in making any sort of progress toward the minimum because of the way in which the Griewangk test problem is littered with multiple local minima which

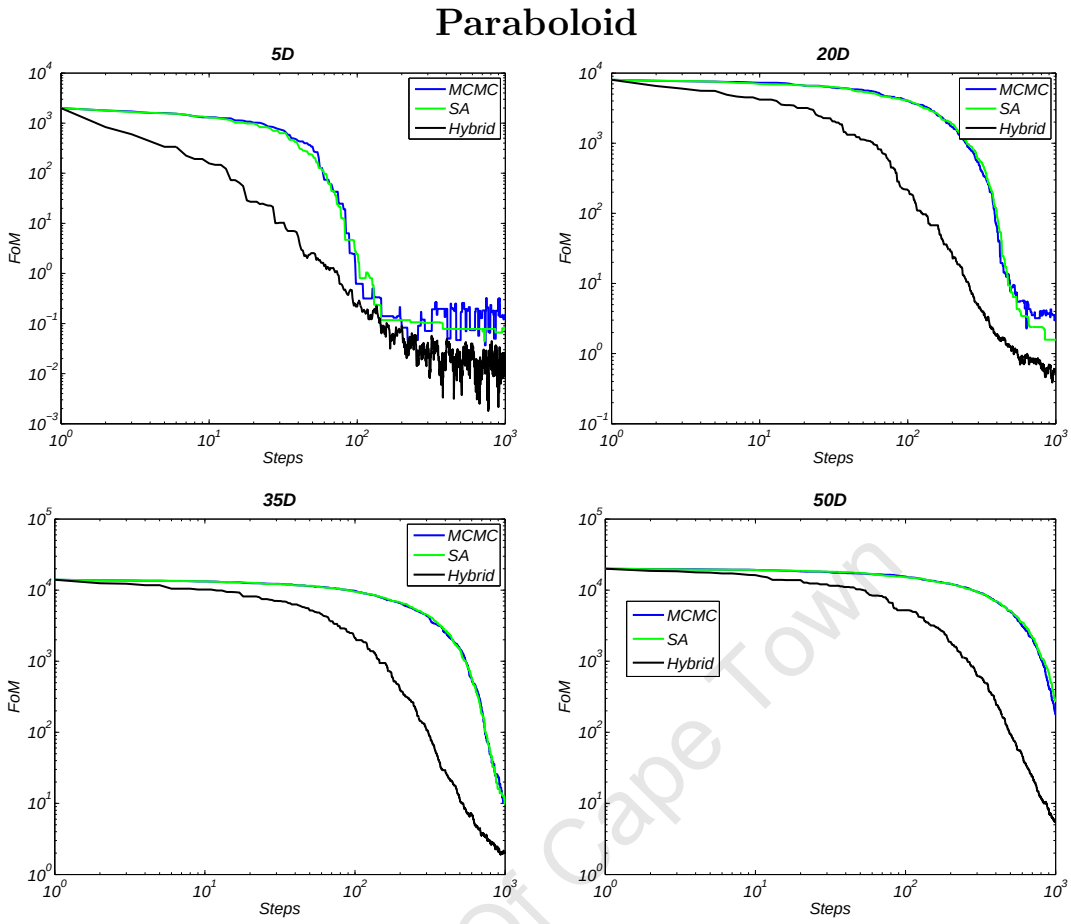


Figure 3.6: **Hybrid** – The Paraboloid test function shows an evenly matched set of results for the 5D case with Hybrid showing a marginally quicker rate of convergence. For higher dimensions Hybrid however shows a more marked improvement both in the rate of convergence and the final FoM after 1000 steps. This is consistently an order of magnitude improvement in the FoM from the standard algorithms of SA and MCMC.

successfully confuse the heuristics for these algorithm.

3.2.3 Varying step size - σ

To address the question of whether MCMC just requires further tuning to get more favourable results we choose a set of varying σ values and compare the results again for the Paraboloid test problems in Figure3.8 while the other two test problems are shown in Appendix A, Figures A.1 and A.2. There are 10 variations considered from $\sigma = 1$ (lightest shade of blue) to $\sigma = 10$ (darkest shade of blue).

A large σ can help improve convergence early on but can also lead to an early saturation of the FoM since the σ is too large to allow the chain to properly navigate

	α	g^{β}	f: If $p > 1$	f: If $p < 1$	Chains	No. Runs
MCMC	1	-	-	-	1	20
SA	1	-	-	-	1	20
Hybrid	1	$g^{0.2}$	$20p + 0.1$	p^2	10	20

Table 3.2: **Paraboloid Function** – This table lists the parameters used for MCMC, SA and Hybrid to produce the results plotted in Figure 3.6. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (20, \ldots, 20)$ for all chains while the range was restricted to $x_i = [-30, 30]$.

	α	g^{β}	f: If $p > 1$	f: If $p < 1$	Chains	No. Runs
MCMC	1	-	-	-	1	20
SA	1	-	-	-	1	20
Hybrid	1	$g^{0.1}$	$15p + 0.2$	$p^{0.2}$	10	20

Table 3.3: **Griewangk Function** – This table lists the parameters used for MCMC, SA and Hybrid to produce the results plotted in Figure 3.7. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (100, \ldots, 100)$ for all chains while the range was restricted to $x_i = [-600, 600]$.

its way to the true minima. This is reflected in the case of the Paraboloid test problem, Figures 3.8, where initially the increase in σ speeds up the rate of convergence but then progressively reaches saturation at higher FoMs as the σ is increased.

A larger σ shows improvement in some instances, in particular the case of the Griewangk test functions. This is to be expected since an increase in σ effectively tunes out the small scale curvature of the minima that MCMC would normally get stuck in and thus 'smooths' the test problem to become a more simple Paraboloid type of function. This is a very specific case of where tuning the σ for MCMC would produce better results but this strategy would generally not yield any benefit.

It is important to highlight that at best the tuned MCMC matches the results of Hybrid and no σ consistently performs well throughout the range of dimensions so potentially a lot of time can be lost in an effort to fine tune the MCMC algorithm. This is particular impractical for higher dimensional problems that require a longer runtime.

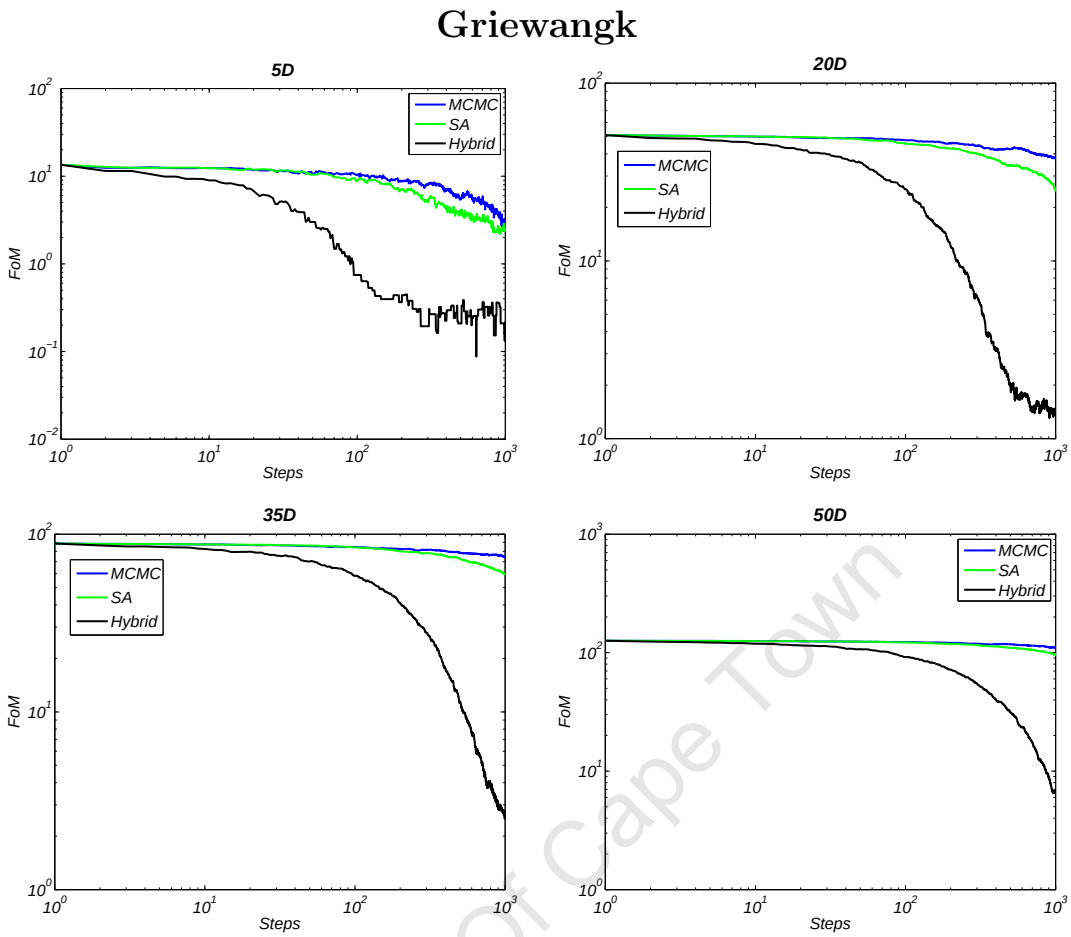


Figure 3.7: **Hybrid** – The Griewangk functions proves particularly testing for SA and MCMC. Hybrid however performs very well in comparison showing a clear improvement in all aspects from 5D-50D. In the 35D case there is an improvement in FoM of close to two orders of magnitude after a 1000 steps.

3.3 Multiple-Try Metropolis Sampling

A tried and tested sampling technique is Metropolis Hastings (MH) [39, 40] which is widely used and recognised as the de facto method when constructing Markov chain algorithms as used in MCMC and SA. This enables a sample to be drawn from a probability distribution but requires the user to supply a proposal distribution. A Gaussian distribution is often used as the default in this instances.

MH is an attractive method to use when little knowledge of the target distribution is available. If one has a good idea of what the target distribution is, then this information can be used to inform the proposal distribution to improve the rate of convergence. In practise this usually means assuming a multivariate normal distribution for both the target and proposal distributions and attempting to estimate the

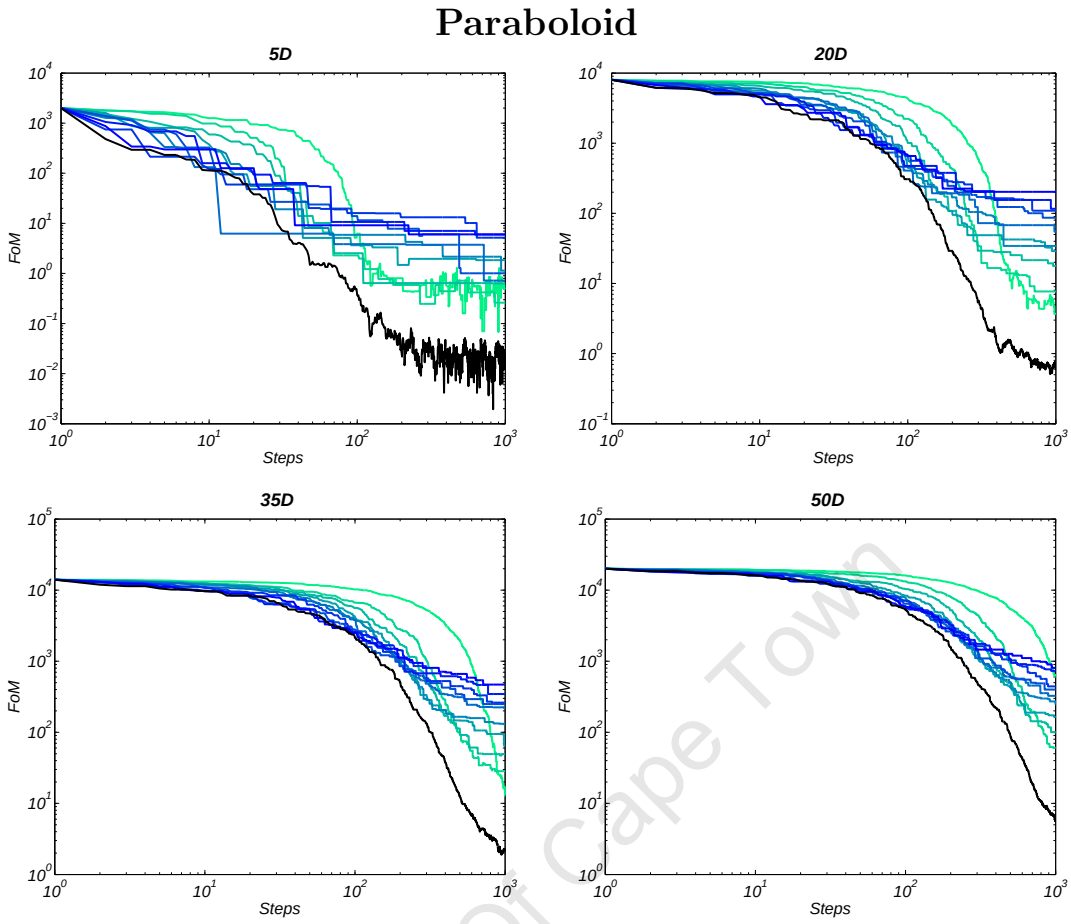


Figure 3.8: **Varying MCMC σ** – The σ is the proposed step size for the MCMC algorithm and is randomly selected from the proposal distribution which is assumed to be a multivariate normal distribution. Illustrated above the σ is varied by a factor from 1 (lightest blue) to 10 (darkest blue) for the MCMC algorithm. In all cases an increase in σ shows a faster initial rate of convergence but results in an earlier saturation at a higher FoM. Hybrid outperforms all variations of MCMC across the dimensions considered.

appropriate covariance matrix for the proposal distribution to best explore the target distribution.

Metropolis Hastings sampling can suffer from a low acceptance rate in the transition probability which translates into a slow rate of progress through the parameter space. In order to improve the acceptance rate and ensure a reasonable convergence time one needs to tune either the size of σ or else the transition probability, usually by introducing and varying a factor α. If for example σ is too large the proposed steps will regularly be rejected and by contrast if σ is too small nearly all the steps will be accepted resulting in a random walk through this space but with a very slow progress towards the true minimum because of the tiny step size.

This problem is compounded in higher dimensions where a well tuned scale parameter is given by, $\sigma/\sqrt{\text{Dim}}$, where Dim is the dimensions, to ensure a reasonable acceptance rate. This results in a predictably slow rate of transversing the parameter space as we consider larger dimensional problems and thus improperly explores the parameters space when we have a limited amount of processing time.

Liu et al. [79] suggested the Multiple-try Metropolis algorithm (MTM) which is a modified MH algorithm that overcomes the above shortfall. The major difference being that multiple proposals are used as opposed to a single proposal per step in the MH algorithm. The obvious benefit of this is that one has a sample range of many positions which allows for a wider set of proposals. This leads to an improved acceptance ratio which is critical to a successfull and speedy convergence. These elements all combine well for application to higher dimensional problems.

MTM is a very subtle sampling technique and in order to introduce the concept in an intuitive way we will address a specific instance of MTM to begin with, followed by the general formalism at the end of this section. The example of MTM given here is described as MTM(II) in Liu et al. [79] and corresponds to an orientation biased Monte Carlo [80].

For simplicity we will consider only a single step of a single chain. MTM begins by selecting k trial proposals, $y_1, \ldots, y_k$ centered around the current position x. In this case a Gaussian distribution is used to select the points. Each trial position is given a weighting, w_n, and a proposed position, Y, is selected from the trial set, $\{y_i\}$ with a probability proportional to the weightings of each of the points in the trial set. A simple case of assigning the weighting is to let it be proportional to the FoM of each positions in the trial set $w_n = \text{FoM}(y_n)$, where n is an index which runs from 1 to k.

The transition probability, which governs how likely the move from the current position (x) to the selected proposed position (Y) is, has to take into account the fact that multiple trial proposals have been made in order to select Y. Given that we have multiple proposed positions we now need an equal number of reference points, $x_1, \ldots, x_k$ from which the selected position Y could have originated. This is addressed by the reference set, $\hat{x}_1, \ldots, \hat{x}_{k-1}$, which samples the points of possible origin and is centered around the proposed position Y and will also include the original starting point $x = \hat{x_k}$. Again a Gaussian distribution was used to select these points.

In MH moving from a bad position to a better position guarantees that the move is accepted; in much the same way MTM wants to ensure that if the proposed move to Y results in a better position from a range of origins, $\hat{x}_1, \ldots, \hat{x}_{k-1}$, then the transition

is guaranteed. This is achieved by taking the ratio of the summed FoMs for the trial set and dividing by the reference set,

$$P(Y|x) = \min\left\{1, \frac{\text{FoM}(y_1) + \cdots + \text{FoM}(y_k)}{\text{FoM}(\hat{x}_1) + \cdots + \text{FoM}(\hat{x}_k)}\right\}. \tag{3.10}$$

MTM is thus focused on how plausible the summed set of proposed points, y_n, is given the set of reference samples, $\hat{x_n}$, centered around the selected proposed position Y, where $n = 1, \ldots, k$.

To generalise the MTM sampling we begin by defining the weighting to be,

$$w(x, y) = \pi(x)T(x, y)\lambda(x, y), \tag{3.11}$$

The simple instance of the weighting would be to set $\lambda \equiv 1$ or alternately one could set,

$$\lambda(x, y) = \left(\frac{T(x, y) + T(y, x)}{2}\right)^{-1}, \tag{3.12}$$

if $T(x, y)$ is symmetric distribution then the weighting reduces to $w(x, y) = \pi(x)$ giving a transition probability of,

$$\min\left\{1, \frac{\pi(y_1) + \cdots + \pi(y_k)}{\pi(\hat{x_1}) + \cdots + \pi(\hat{x_k})}\right\} \tag{3.13}$$

which is equivalent to the example introduced in the beginning of this section where $\pi(x) \equiv \text{FoM}(x)$ and thus producing Eq. (3.10).

The general steps followed when considering an arbitrary weighting, w, and distribution, T, for MTM are:

1. Draw a set of k trial proposals $y_1, \ldots, y_k$ from $T(x, \cdot)$. Compute the weightings $w(y_j, x)$ for $j = 1, \ldots, k$

2. Select Y from the set $\{y_1, \ldots, y_k\}$ with a probability proportional to the weightings given by Eq. (3.11).

3. Produce a reference set of $\{\hat{x_1}, \ldots, \hat{x_{k-1}}\}$ from the distribution $T(y, \cdot)$ and set $\hat{x_k} = x$ to be the same as the current position under consideration.

4. Accept the step to Y with a probability proportional to

$$P(Y|x) = \min\left\{1, \frac{w(y_1, x) + \cdots + w(y_k, x)}{w(\hat{x_1}, Y) + \cdots + w(\hat{x_k}, Y)}\right\}. \tag{3.14}$$

   and reject it with probability 1-$P(Y|x)$.

5. If the step to Y is rejected then remain at the current position, x.

3.3.1 MTM Results

In this section we evaluate how much of an impact the MTM sampling has when combined with the MCMC and Hybrid algorithms. This is compared with the Hybrid algorithm which convincingly outperformed SA and MCMC in the previous section. The results are illustrated in Figure 3.9 , 3.10 and 3.11 using the parameters listed in Table 3.4, 3.5 and 3.6. The same methodology is employed as with all the other plots where a total of 50 runs is done on each algorithm on each test problem. The lowest FoM at each step is recorded and plotted, showing the best results over all the runs. Throughout the range of benchmark problems Hybrid-MTM outperforms its optimizing counterparts.

In the case of the Rosenbrock test probelm, shown in Figure 3.9, Hybrid and Hybrid-MTM produce similar results. MCMC-MTM saturates in each of the dimensional cases at less than 100 steps and then doesn't match the final results of Hybrid and Hybrid-MTM.

	α	g^{β}	$f(p>1)$	$f(p<1)$	Chains	No. Runs	k
MCMC-MTM	1	-	-	-	1	20	10
Hybrid-MTM	0.1	$g^{0.6}$	$1.5p+0.5$	p^2	10	20	10

Table 3.4: **Rosenbrock Function** – This table lists the parameters used for MCMC-MTM and Hybrid-MTM to produce the results plotted in Figure 3.9 while the parameters for MCMC remain the same as given in Table 3.1. The parameter k is the number of multiple trial proposals used per chain. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (-2, \ldots, -2)$ for all chains while the range was restricted to $x_i = [-2.048, 2.048]$.

In Figure 3.10 the Paraboloid benchmark functions is shown. Here Hybrid-MTM outperforms all other algorithms convincingly showing a final FoM in excess of seven orders of magnitude better than MCMC-MTM for all dimensions. Only in the lower dimensional case (5D) does Hybrid initially outperform Hybrid-MTM but cannot match the final FoM produced by Hybrid-MTM. MCMC-MTM also shows gains in its rate of convergence in higher dimensions but again saturates marginally above the final FoM of Hybrid.

The final test problem's results for the Griewangk function is depicted in Figure 3.11. This again shows Hybrid-MTM producing the best FoM across the board

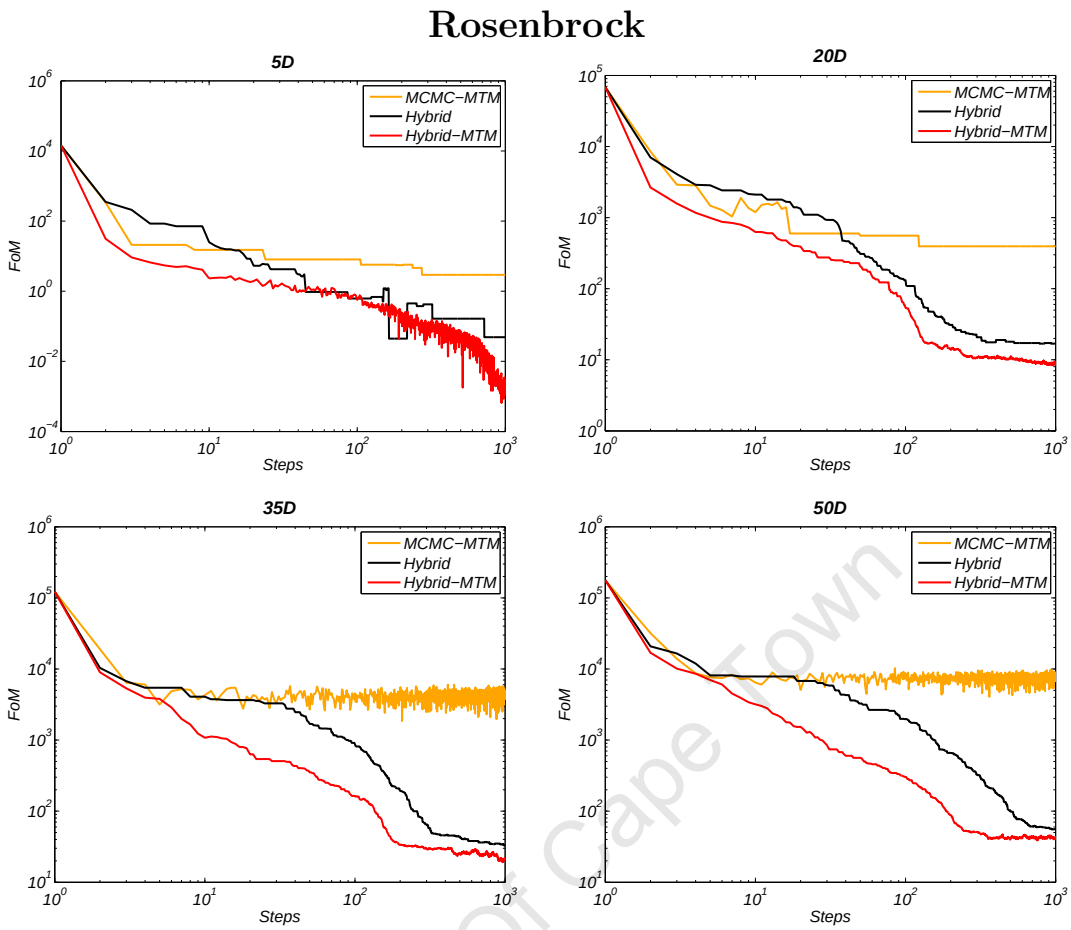


Figure 3.9: **Hybrid-MTM Results** – The Rosenbrock functions produces similar results for Hybrid and Hybrid-MTM from 5D-50D. Hybrid-MTM does generally show a better rate of convergence and a small but noticeable improvement of the FoM in all dimensions. MCMC-MTM initially matches the Hybrid algorithms but appears to saturate early on, steps < 100, and then never converges to the true minima.

with Hybrid only matching the results for the 5D case. The difference in the final FoM for Hybrid-MTM is again impressive with a eight order of magnitude improvement shown in the 20D case while a six orders of magnitude improvement is achieved for the 35D and 50D cases. The MCMC-MTM algorithm shows that it is better suited to this test problem since it outperforms Hybrid in all the dimensions greater than 5D.

MTM sampling is more complex and evaluates many more points than the standard MH sampling technique. This would imply that it is a more computationally intensive method to employ. This may in turn suggest that using the FoM as a function of steps is not a fair comparison with the simpler and naively faster algorithms of standard SA and MCMC. To examine this we generate a set of plots, much the same

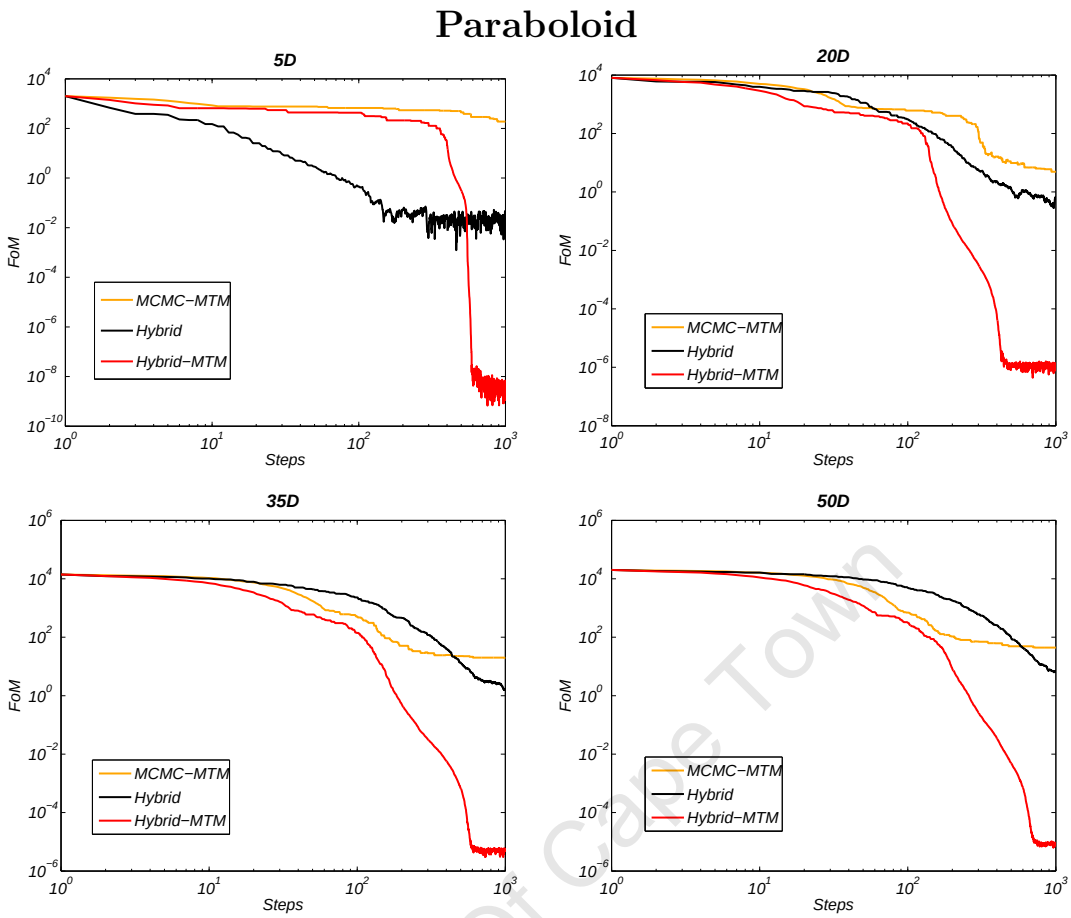


Figure 3.10: **Hybrid-MTM Results** – Hybrid-MTM outperforms both MCMC-MTM and Hybrid for all dimensions with at least a 10^5 times improvement in FoM compared with its nearest rival. The rate of improvement is more prominent in the higher dimensional cases. Only in the 5D instance does Hybrid initially converge to a better FoM at a lower number of steps than Hybrid-MTM which ultimately produces a much better FoM than Hybrid after 400 steps. MCMC-MTM also shows a faster rate of convergence than Hybrid in the higher dimensional cases ($>$ 20D) but doesn't match Hybrid with its final FoM after 1000 steps. These results are for the best FoM for each algorithm over 50 runs.

as previously presented, applying all the algorithms to the three benchmark problems across a range of dimensions. In this case however we compare the best FoM for each algorithm at every step as a function of computational time, given in seconds. These results are shown for the Paraboloid function in Figure 3.12 while the remaining two test functions are shown in Appendix A, Figures A.4 and A.4.

It can be seen from the Paraboloid test problem shown in Figure 3.12 that Hybrid-MTM, despite its added complexity and computational expense, still manages to outperform the other simpler algorithms. This is due to two factors, one being that

	α	g^{β}	$f(p>1)$	$f(p<1)$	Chains	No. Runs	k
MCMC-MTM	1	-	-	-	1	20	10
Hybrid-MTM	0.1	$g^{0.2}$	$2.5p + 1e^{-3}$	p^2	10	20	10

Table 3.5: **Paraboloid Function** – This table lists the parameters used for MCMC-MTM and Hybrid-MTM to produce the results plotted in Figure 3.10 while the parameters for MCMC remain the same as given in Table 3.2. The parameter k is the number of multiple trial proposals used per chain. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (20, \ldots, 20)$ for all chains while the range was restricted to $x_i = [-30, 30]$.

	α	g^{β}	$f(p>1)$	$f(p<1)$	Chains	No. Runs	k
MCMC-MTM	1	-	-	-	1	20	10
Hybrid-MTM	0.5	$g^{0.6}$	$5p + 5e^{-3}$	p^2	10	20	10

Table 3.6: **Griewangk Function** – This table lists the parameters used for MCMC-MTM and Hybrid-MTM to produce the results plotted in Figure 3.11 while the parameters for MCMC remain the same as given in Table 3.3. The parameter k is the number of multiple trial proposals used per chain. The starting point in each instance for this test function was $(x_1, \ldots, x_i) = (100, \ldots, 100)$ for all chains while the range was restricted to $x_i = [-600, 600]$.

it has such a rapid rate of convergence to very good FoM values. This becomes particularly more evident as we consider higher dimensional cases. The MCMC-MTM shows signs of faster convergence, again in higher dimensions, but suffers from saturating at a high FoM leaving it unable to compete with Hybrid-MTM.

The second favourable aspect of MTM is that it has a high acceptance rate even in large dimensional problems. It thus successfully averts the problems faced by SA and MCMC using standard MH sampling discussed in Section 3.3. This factor combines particularly well with Hybrid's methodology which allows for a range of step sizes to be explored in the parameter space and the best option sought out at every step.

The standard algorithms of SA, MCMC and even MCMC-MTM are capable of taking more steps in the same length of time as Hybrid or Hybrid-MTM but this doesn't offer any gains since both Hybrid and Hybrid-MTM still converge to lower FoM values, even when considering CPU running time. In Figure 3.12 SA, MCMC and MCMC-MTM are all being run for 3000 steps while Hybrid and Hybrid-MTM are only being run for 1000 steps. The resulting time taken for each algorithm is then

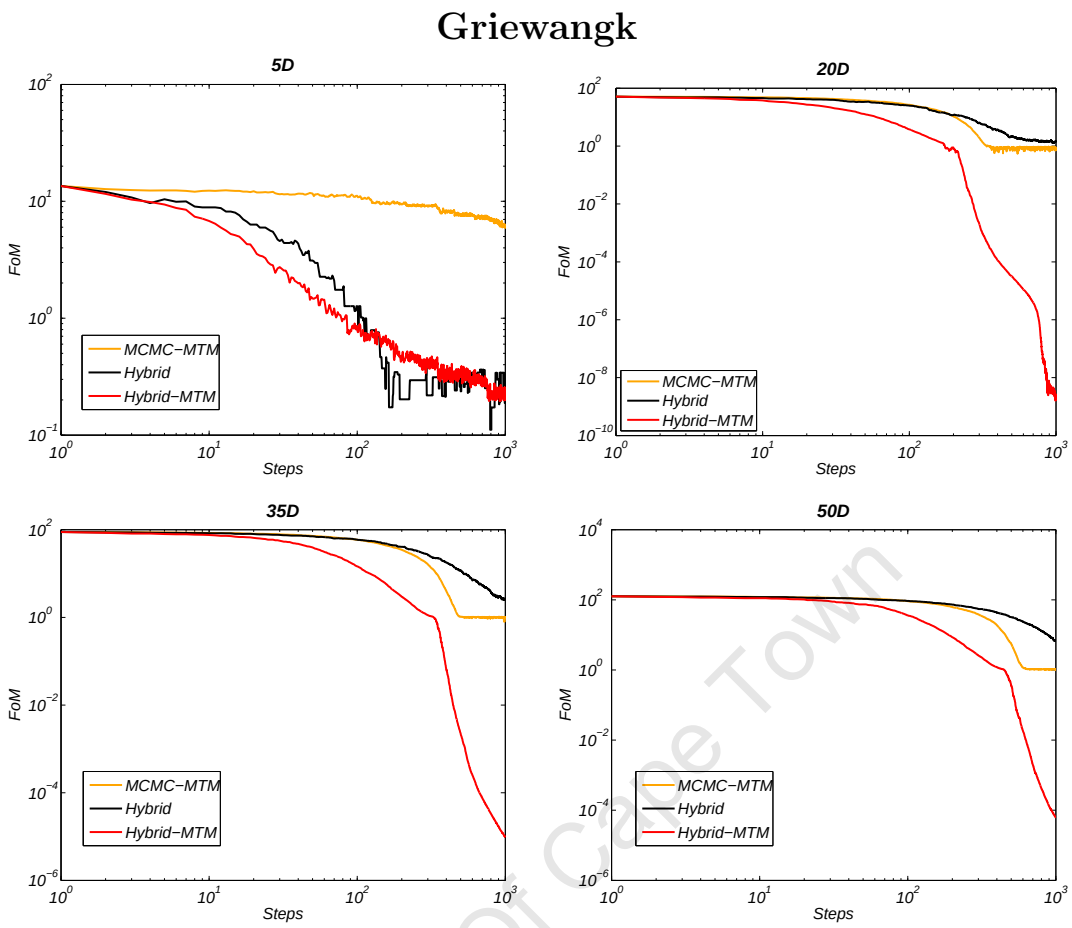


Figure 3.11: **Hybrid-MTM Results** – The Griewangk test problem produces results very similar to the Paraboloid case but with a more dramatic disparity in the rate of convergence where Hybrid-MTM is far more rapid compared to the other two algorithms. Here a $> 10^8$ times improvement is noted in FoM compared to both MCMC-MTM and Hybrid in the 20D case. Only in the 5D example is there a similar rate of convergence and final FoM for both Hybrid and Hybrid-MTM. MCMC-MTM also shows in the 35D and 50D cases a rapid convergence at close to 1000 steps an FoM ~ 0 which then outperforms Hybrid in these two instances.

roughly equivalent but the results still drastically favour Hybrid-MTM and to a lesser extent Hybrid.

3.4 Conclusions

In this chapter we built on the concepts of non-linear optimization along with the the commonly used optimization techniques of PSO, SA and MCMC which were introduced in Section 1.2.

A new optimization heuristic named Hybrid was reviewed and successfully illus-

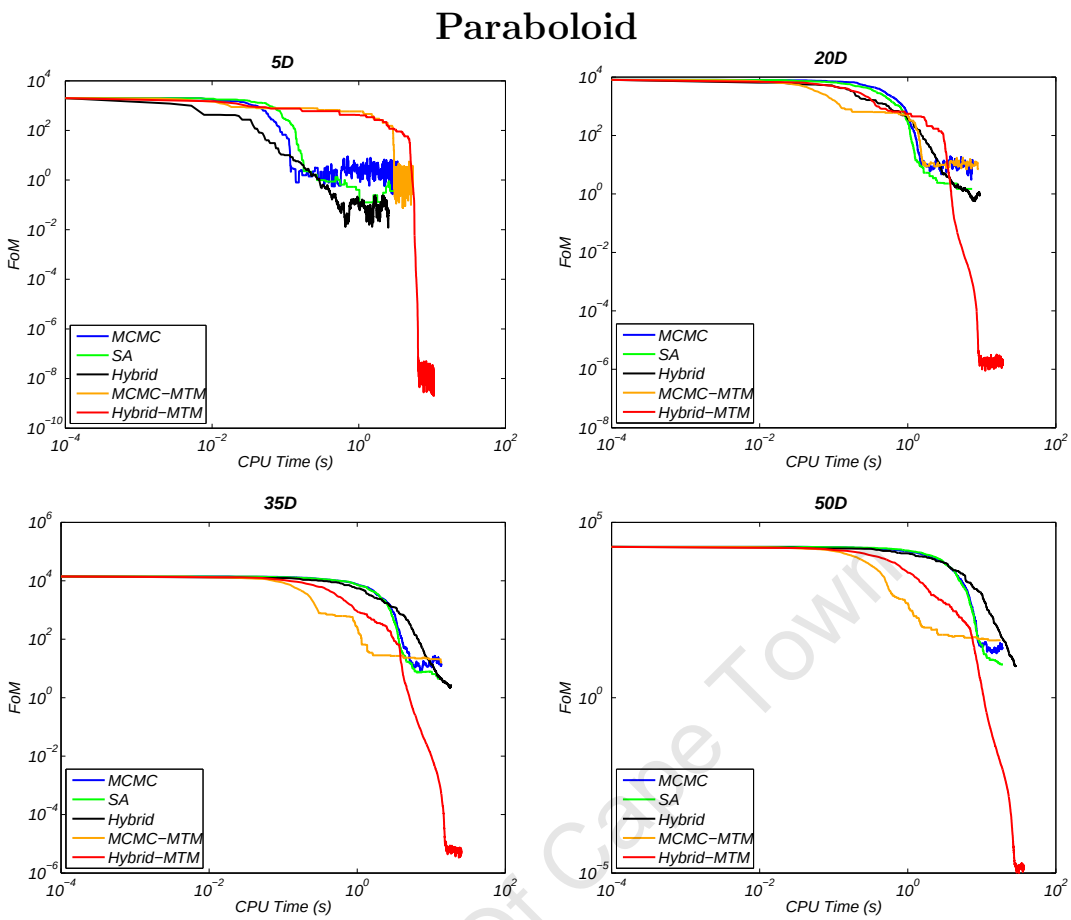


Figure 3.12: **CPU Time** – Here a comparison between MCMC, SA, Hybrid, MCMC-MTM and Hybrid-MTM is done on the Paraboloid test problem to record how the FoM changes as a function of CPU time recorded in seconds. As higher dimesions are considered so Hybrid-MTM's performance improves compared to the other algorithms. Hybrid-MTM, although beaten initially (< 1 sec) by the other algorithms, rapidly converges to a very good FoM showing excellent performance as a function of time. MCMC-MTM again also shows that its is well suited to higher dimensional problems with good convergence to low FoM initially but then reaches a level of saturation too early to compete with Hybrid-MTM.

trated how combining aspects of both PSO and SA helped it emerge as a more efficient optimization algorithm. This was highlighted by using multi-dimensional versions of the Rosenbrock, Paraboloid and the Griewangk functions as test problems. The dimensions used were 5D, 20D, 35D and 50D. The three algorithms of Hybrid, SA and MCMC were then applied to these multi-dimensional test problems. In each case Hybrid showed a marked improvement in the rate of convergence with a significantly lower FoM after 1000 steps. The distinction between Hybrid, SA and MCMC became particularly clear in the case of the higher dimensional test functions.

In an attempt to gain equal performance out of MCMC to match that of Hybrid, the step size standard deviation, σ, is tuned across a range from $\sigma = 1$ to $\sigma = 10$ for each of the test problems. This was found to have mixed results and did not warrant the extra time needed to successfully tune σ. Even in the event that a good σ was found, this would not be applicable across the range of dimensions chosen and would have to be individually tuned for each dimensional case. The tuning is time consuming, in particular for higher dimensional cases, and does not match Hybrid's more robust ability to effectively handle a range of multi-dimensional problems.

Lastly an advanced sampling technique was introduced and reviewed, Multiple-Try Metropolis (MTM). This was first used in conjunction with MCMC and showed great promise. When it was combined with Hybrid drastic improvements were noted. These were particularly evident in higher dimensions where instances of outperforming standard methods by eight orders of magnitude were often encountered. The computational efficiency of this new sampling method coupled with Hybrid was also examined by comparing the CPU time on each of the test problems with that of all the other optimization algorithms. This rather surprisingly showed that in higher dimensional cases, despite the more complex algorithmic calculations required by MTM, it more quickly (in CPU time) converged to a lower minimum than its simpler optimizing counterparts. This can be attributed to the higher acceptance ratio that MTM sampling maintains in higher dimensional problems which combines well with Hybrid's ability to explore a wide range of step sizes and thus allows for greater gains in finding the minimum effectively in CPU time relative to standard techniques.

In this thesis we have re-explored the MTM and Hybrid methods and, for the first time, combined them, yielding a new algorithm, Hybrid-MTM, that significantly outperforms both its parents and all other tested optimisation algorithms, especially in high-dimensional problems.

Chapter 4

Constraints on Dark Energy at low z from BBN

This chapter is partly composed of the work presented in the published paper [1] while the counter example presented in Section 4.5 is an extension of this work.

In a general dark energy model the dynamics at late times are completely independent of the dynamics of the early universe, in other words the values of $w(z)$ at small and large z can be completely uncorrelated. However in this chapter we will show that for a popular class of dark energy models known as tracking models (also referred to as scaling models), there are indeed strong implications for late time dynamics given early universe constraints arising from Big Bang Nucleosynthesis (BBN) and the Cosmic Microwave Background (CMB).

Both BBN and the CMB limit the amount of dark energy at those respective epochs to be less than about 5% of the total energy density of the universe [71,76]. In tracking models, as discussed in Section 2.5.2, the equation of state of the dark energy, $w(z)$, follows the dominant component at that period. In other words $w(z) = 1/3$ during radiation domination and $w(z) = 0$ during matter domination. As a result the fraction of dark energy, $\Omega_{\rm DE}(z)$, is constant in time. This means that any constraint that applies to dark energy at early times will also apply subsequently while the field is still tracking and in particular up to some redshift z_t at which the field breaks away from tracking and dark energy becomes dominant, initiating cosmic acceleration.

If we allow only a simple transition from tracking to dark energy domination then we will show that in these tracking models BBN and CMB constraints imply powerful limitations on the range of allowed dynamics at $z < 1$ which will make detecting dark energy dynamics of such models very difficult and will perhaps have to wait for the Stage-IV dark energy experiments such as SKA and JDEM. One natural issue that

could be raised with the generality of these results is that the transition from tracking to dark energy domination might not be simple and may have significant substructure, e.g. the transition from $w = 0$ to $w \simeq -1$ could proceed via two stages with an intermediate plateau at w_l. Although fine tuned and somewhat unnatural such a model will allow strong dynamics at $z < 1$ while still meeting the BBN constraints. However we show in the second part of this chapter that such models are unlikely to be compatible with the WMAP [11, 12] angular power spectrum for the CMB. Our main conclusion therefore is that it is difficult or perhaps impossible to produce tracking dark energy models that meet all observational constraints and also have dynamics which will be detectable before the Stage-IV dark energy experiments.

4.1 Recipe for Solving z_t

Using the BBN constraint where $\Omega_{\rm DE} \leq \epsilon \equiv 0.045$ [71] and assuming a tracking scalar field until the transition from matter dominance to DE dominance, we are able to calculate the transition redshift, z_t, at which the scaling ceases. In this section we introduce a recipe for determining z_t given a general parameterisation of the dark energy equation of state, $w(z)$, which is specified after the field breaks away and dark energy becomes dominant (when $z \leq z_t$). This formalism is used in the next section where a warm up problem is introduced and subsequently in Section 4.5 detailing a proposed counter example.

We want a prescription that allows us to solve for z_t given a specified $w(z)$ when $0 \leq z \leq z_t$. In order to relate the BBN constraint $\Omega_{\rm DE} \leq \epsilon$ to $w(z)$ we require an intermediate step of relating ϵ to $f(z)$, given in Eq. 4.3. First we consider the expression for the ratio of dark energy density (neglecting radiation),

$$\begin{aligned} \Omega_{DE}(z) &\equiv \frac{\rho_{DE}(z)}{\rho_{cr}(z)} = \epsilon, \\ &= \frac{\Omega_{\rm DE} f(z)}{\Omega_{\rm DE} f(z) + \Omega_m (1+z)^3}, \end{aligned} \tag{4.1}$$

where the universe is assumed to be flat and composed of matter and dark energy with densities given today by $\Omega_{\rm DE} \sim 0.7$ and $\Omega_m \sim 0.3$ while Ω_r makes a negligible contribution and is thus omitted for simplicity.

Applying the upper limit of the BBN constraint at transition between matter domination and dark energy dominance produces $\Omega_{DE}(z_t) = \epsilon$ and solving for $f(z_t)$

produces a characteristic relationship that will be repeatedly used,

$$\begin{aligned} \frac{f(z_t)}{f(z_t)+\frac{1}{r}(1+z_t)^3} &= \epsilon, \\ f(z_t)(1-\epsilon) &= \frac{\epsilon}{r}(1+z_t)^3, \\ f(z_t) &= \frac{\epsilon}{r(1-\epsilon)}(1+z_t)^3, \end{aligned} \tag{4.2}$$

where $r = \Omega_{\rm DE}/\Omega_m$. The evolution of dark energy is given by the left side of the equation and is defined as

$$f(z) \equiv \exp\left[3\int_0^z \frac{1+w(z')}{1+z'}dz'\right], \tag{4.3}$$

and allows a specific form of $w(z)$ to be considered between $z = z_t$ and $z = 0$. By solving the expression for $f(z_t)$ from Eq.(4.3) and then substituting it into Eq.(4.2) we are able to solve for z_t.

4.2 Linear Parameterisation

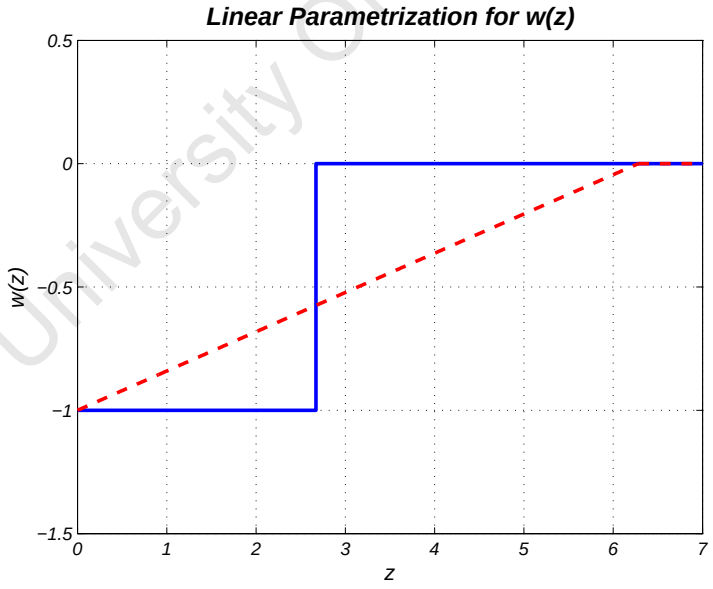


Figure 4.1: **Linear Parameterisation** – Showing the step function (blue line) and straight line (red dashed line) parameterisation for the dark energy equation of state, $w(z)$, as a function of redshift, z. In combination with the BBN constraint and assuming a tracking model the step function parameterisation has a $z_t = 2.67$ while the straight line parameterisation has a higher transition at $z_t = 6.28$.

Next we consider a warm up problem were we examine what happens after dark energy stops tracking. We can assume a transition in the equation of state at $z = z_t$ which can take any form of parameterisation to represent the transition but for simplicity we only consider a step-function and a linear transition from $w(z = 0) = -1$ to $w(z_t) = 0$.

Although the step-function parameterisation, shown in Figure 4.1, is rather unphysical it has the benefit of significantly simplifying our calculation. Specifically when we consider how the integral in Eq.(4.3) reduces to,

$$f(z_t) = (1 + z_t)^{3(1+w_{\rm DE})}. \tag{4.4}$$

Which in turn makes it easy to solve for $w_{\rm DE}$ by using Eq.(4.4) in Eq.(4.2),

$$w_{\rm DE} = \frac{\ln\left[\frac{\epsilon}{r(1-\epsilon)}\right]}{3\ln(1+z_t)}, \tag{4.5}$$

or alternately solving in terms of z_t,

$$z_t = \left[\frac{\epsilon}{r(1-\epsilon)}\right]^{1/(3w_{\rm DE})} - 1. \tag{4.6}$$

The best current data suggests a $w_{\rm DE} \sim -1$ [81] which when substituted into the above equation results in a solution of $z_t = 2.67$ where .

Since a step function represents an extremely rapid transition in the parameterisation of the equation of state, a simple straight line between $z = z_t$ and $z = 0$, shown in Figure 4.1, can be used to evaluate a more gradual transition from matter to dark energy domination.

Following the same logic as we did with deriving the z_t above, we first establish the form of the integral in Eq.(4.3) for a straight line $w(z) = w_0 + w_1 z$ where $w_0 = -1$ since we assume for simplicity that $w(z = 0) = -1$. This implies that,

$$f(z_t) = \exp\left[3w_1(z_t - \ln(1 + z_t))\right], \tag{4.7}$$

and by substituting Eq.(4.7) into Eq.(4.2) we can solve for w_1,

$$w_1 = \frac{\ln\left[\frac{\epsilon}{r(1-\epsilon)}(1+z_t)^3\right]}{3\left[z_t - \ln(1+z_t)\right]}. \tag{4.8}$$

If we substitute w_1 from the above equation into $w(z_t) = w_0 + w_1 z_t = 0$ and numerically solve for z_t a result of $z_t = 6.28$ is recovered.

In Figure 4.1 we can note a contrast between the sharp transition of the step function and the more gradual transition for the linear parameterisation producing a z_t which is much higher for the gradual transition than that of the sharper transition. This suggests that in general the z_t is forced to become higher from the BBN constraint as we consider a parameterisation that allows for a more gradual transition. From the above examples, one can deduce that the smoother or more gradual the transition from matter to DE domination, the earlier the corresponding transition redshift takes place, resulting in a larger z_t. This can be understood since a slower change in $w(z)$ results in a longer time to change $\Omega_{DE} = 0.045$ to $\Omega_{\rm DE} = 0.7$. This is in contrast to a more rapid transition which would result in a much later or lower z_t. By this reasoning we can conclude that a step function will have the lowest allowed z_t for any parameterisation where an equation of state $w \geq -1$ is being used.

4.3 Double Exponential Potential

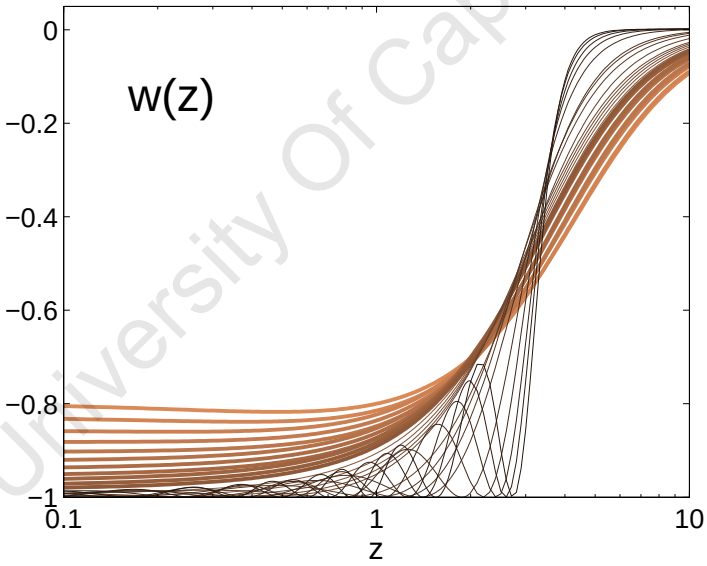


Figure 4.2: ***w(z)* for Double Exponential Potential [1]** – This is for values $-30 \leq \mu \leq 1$ used in Eq. (4.9). The light brown curves correspond to values from $\mu = 1$ (for which $w(0) = 0.8$) to $\mu = 0, w(0) = 1$. The dark brown curves are for a range of negative values of μ which produce a very rapid transition and can be identified by their characteristic oscillatory nature.

Up until now we have considered parameterisations which are theoretical toy models and rather unphysical in nature so we now turn our attention to more realistic models that could give us clues about the feasibility of detection.

As introduced in Section 2.5.2, a single exponential potential is well known to give early scaling [69, 70, 82] but cannot also lead to late-time acceleration. One well-studied way to combine the two is via the double exponential potential, as considered by Barreiro, Copeland and Nunes [72]:

$$V(\phi) = M_1^4 e^{-\lambda\kappa\phi} + M_2^4 e^{-\mu\kappa\phi}. \tag{4.9}$$

This potential with the correctly tuned values can lead to the desired scenario where the universe goes through radiation and matter dominated epochs, during which the field scales with the background fluid, and importantly, at late times, evolves into the scalar field dominated regime. Such a potential is theoretically well motivated, because it is expected to arise as a result of compactifications in superstring models [82].

Unlike a perfectly scaling tracking model, as we have assumed to this point, $\Omega_{\rm DE}$ actually decreases in the transition to matter domination and we have $\Omega_{\rm DE} < 3/4 \times \epsilon$ during matter domination. Here we explain how this happens and how it leads to this even more stringent result. Assuming that the scalar field is already in the scaling regime at the epoch of nucleosynthesis (which occurs during the radiation dominated era), using $\Omega_{\rm DE} = n/\lambda^2$ along with the BBN constraint $\Omega_{\rm DE} < \epsilon$ implies a constraint on the value of λ [82]

$$\lambda^2 > \frac{4}{\epsilon} \quad \Rightarrow \quad \lambda \gtrsim 9.43, \tag{4.10}$$

where $n = 4$ is used for radiation and $n = 3$ for matter domination. Now that we have fixed the value for λ in terms of ϵ we can calculate what the constraint on the matter dominated epoch will be,

$$\begin{aligned} \Omega_{\rm DE} &= \frac{3}{\lambda^2} < \frac{3}{4/\epsilon}, \\ &< \frac{3}{4}\epsilon. \end{aligned} \tag{4.11}$$

If the BBN constraint in Eq.(4.10) is satisfied we will have tracking during radiation and matter dominated eras, and we need $\mu < \sqrt{2}$ if we want the field to leave the scaling regime at late times and give rise to an accelerated expansion. In addition we must choose M_2 such that $\Omega_{\rm DE} \sim 0.7$ and $\Omega_m \sim 0.3$ today (which implies $M_2 \sim 10^{-31} m_{\rm pl}$ for $\mu \sim 1$). A wide range of initial conditions were shown to be compatible with this scenario [72].

We numerically solve the evolution equations for a scalar field with the potential

in Eq. (4.9) and for radiation and matter fluids,

$$\ddot{\phi} + 3H\dot{\phi} + V_{,\phi} = 0, \tag{4.12}$$

$$\dot{\rho}_r + 4H\rho_r = 0, \tag{4.13}$$

$$\dot{\rho}_m + 3H\rho_m = 0, \tag{4.14}$$

$$H^2 = \frac{\kappa^2}{3}\left[\frac{1}{2}\dot{\phi}^2 + V(\phi) + \rho_m + \rho_r\right]. \tag{4.15}$$

In order to maximise deviation from ΛCDM we choose the value of λ to be the upper bound given in Eq.(4.10) and $M_1 = 10^{-14}$ where we use Plank units and set $\kappa = 1$.

The field will scale during the radiation dominated era, with the equation of state of 1/3, and then proceed to scale with matter when the universe enters in the matter dominated regime, with an equation of state of zero, and will finally dominate the energy density of the universe at late times. The shape of the curve for $w(z)$ as it transitions from zero to the final value today will depend on the value of the parameter μ [72], this is depicted in Figure 4.2.

There are two cases that are considered, namely a negative μ where $-30 \leq \mu < 0$ (dark brown lines in Fig. 4.2) and a positive μ where $0 \leq \mu \leq 1$ (light brown lines in Fig. 4.2). In the negative case the field oscillates around the minimum of the potential and is clearly illustrated in Figure 4.2. The potential in this case has a global minimum and for $z \lesssim 0.2$ the equation of state satisfies $w \leq -0.98$. Another clear feature of the negative μ case is that there is a rapid transition from $w = 0$ to $w \sim -1$ with only a slight variation due to the oscillations. This results in relatively low z_t values since we are approximating the case of the step function described in our warm up problem.

In the second case we consider positive values of μ. This has the effect of modifying the slope of the potential which allows for a smoother form of $w(z)$. It also has the benefit of being able to yield values of $w(z)$ significantly different from -1 today, e.g. for $\mu \sim 1$ will result in $w(0) \sim -0.8$ which is consistent (at about the 2σ level [53,83]) with current observations and which we therefore take as the upper bound for μ. Here again we note a similarity of what we encountered with our toy model of the gradual linear transition for w. The more gradual the transitions, the higher the resulting z_t, in order to ensure that the $\Omega_{\rm DE} \sim 0.7$ today. This is compounded in the instance where the models considered only reach a minimum of $w(0) \sim -0.8$ which will require an even longer period at a negative equation of state to satisfy the $\Omega_{\rm DE} \sim 0.7$ condition. Figure 4.3 illustrates this point and one can see that the positive μ models (light brown lines) start increasing from $\Omega_{\rm DE} = \epsilon$ at higher z values

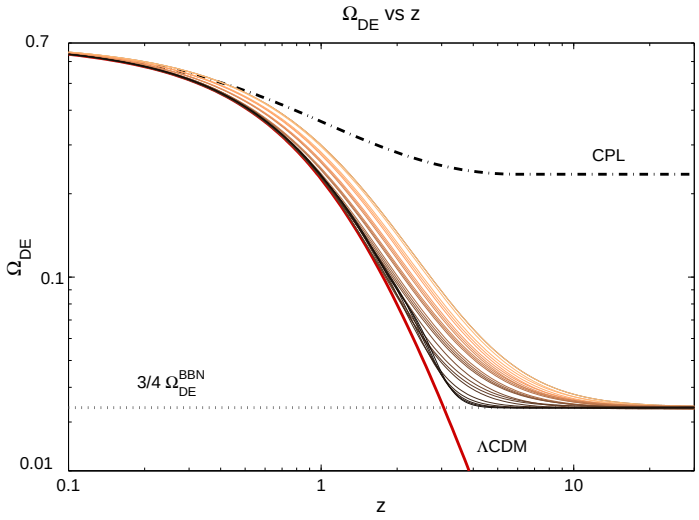


Figure 4.3: **Evolution of $\Omega_{DE}(z)$ adapted from [1]** – for the double exponential potential with $-30 \leq \mu \leq 1$, light brown lines for positive values of μ and dark brown lines for negative values of μ. This illustrates how $\Omega_{\rm DE}$ increases for different values of μ from the BBN limits of $3/4\epsilon = \Omega_{\rm DE}(z = z_{\rm BBN}) = 3/4 \times 0.045$ to $\Omega_{\rm DE}(z = 0) = 0.7$ today. For comparison we also show the curves for Λ and the CPL [8,9] $w(z)$ with the lowest asymptotic value of $\Omega_{\rm DE}$ in this model while still assuming $w \geq 1$, showing its inability to match the BBN constraint and describe a canonical scalar field. Figure 4.4 shows the corresponding observational quantities for the scaling quintessence models.

(earlier) since they don't have the same steep increase for $\Omega_{\rm DE}$ at low redshifts as the negative μ models have (dark brown lines).

4.3.1 Observational Implications

We now address the prospect of observational detection of dark energy dynamics for the specific case of the double exponential potential introduced in the previous section. We employ two methods to achieve this, illustrated in Figure 4.4. The first figure in the top panel shows the difference in distance modulus $\Delta\mu \equiv \mu_{\rm DE}(z) - \mu_{\Lambda}(z)$ given by:

$$\Delta\mu = 5\log_{10}\left(\frac{d_{L,\rm DE}(z)}{d_{L,\Lambda(z)}}\right), \tag{4.16}$$

where $d_L(z)$ is the corresponding model luminosity distance, given by (for a flat universe),

$$d_L(z) = \frac{c}{H_0}(1+z)\int_0^z \frac{dz'}{E(z')}, \tag{4.17}$$

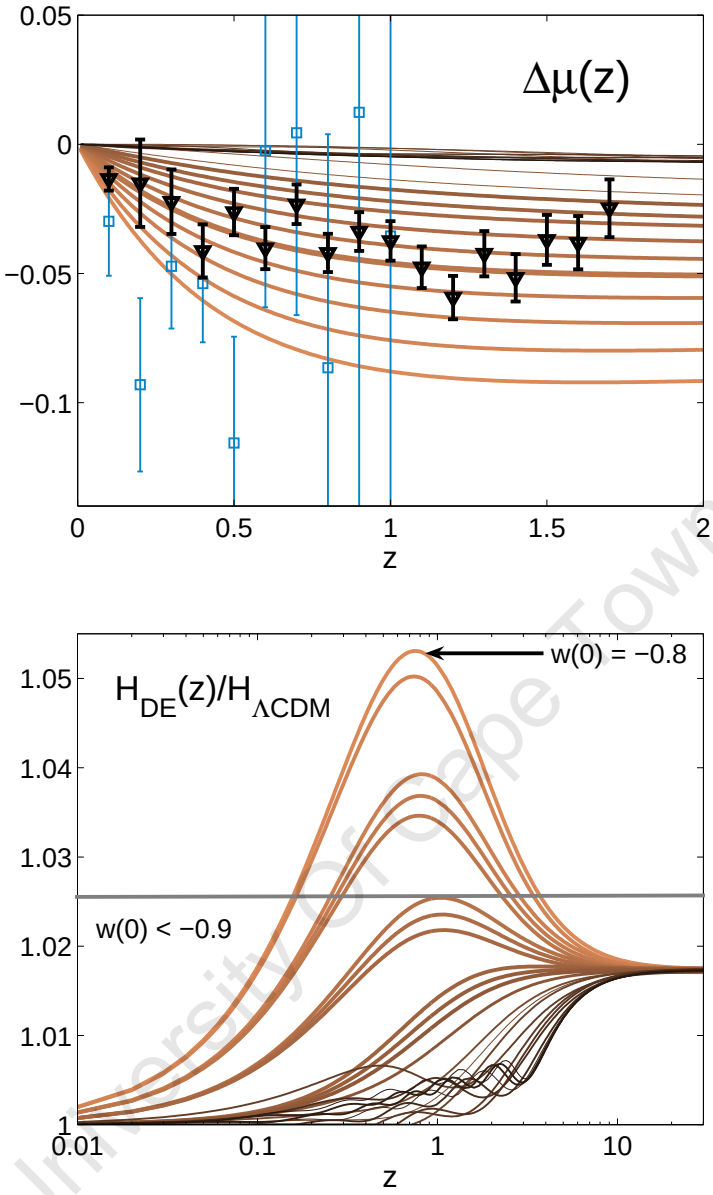


Figure 4.4: **Observable results for Double Exponential Potential [1]** – This figure shows a range of $-30 \leq \mu \leq 1$ values considered for the double exponential potential producing the results for the $\Delta\mu(z)$ (top panel) and the ratio $H_{\rm DE}/H_{\Lambda{\rm CDM}}$ (bottom panel). The same convention is used as in previous plots of the double exponential potential where the negative μ (light brown lines) and positive μ (dark brown lines) values considered are clearly distinguished. The error bars in the top panel correspond to simulated data for DETF Stage-III (large boxed errors) and Stage-IV (small triangular errors) supernova surveys respectively. The Stage-III errors are produced for the bottom line, $w(z = -0.8)$, while the Stage-IV (SNAP-like) errors are produced for the corresponding $w(0) = 0.9$ model (thicker line asymptoting to ~ 0.04 mag). The notable improvement in the error bars is as a result of no systematic errors being expected for Stage-IV surveys.

and $E(z)$ is specified by,

$$E(z) = \sqrt{\Omega_m(1+z)^3 + \Omega_{\rm DE} f(z) + \Omega_{\rm rad}(1+z)^4}, \tag{4.18}$$

and $f(z)$ is given by Eq. (4.3) where the form of $w(z)$ being used can be entered. While the second figure on the bottom panel is a ratio of $H_{\rm DE}/H_\Lambda$ where each $H_{\rm DE}$ was calculated numerically from the range of μ values described in the previous section.

Again we considered the two cases of a negative μ where $-30 \leq \mu < 0$ and a positive μ where $0 \leq \mu \leq 1$. The negative μ case, as shown by the dark lines in Figure 4.3, display only a very small deviations from ΛCDM at low redshift. This implies that all the negative μ models also only show a fractional detectability resulting in 1.5% for $H(z)$ and less than 0.015 mag for $\Delta\mu$ shown in Figure 4.4. In both panels of Figure 4.4 it is clear that the negative μ results are the most challenging to distinguish from standard ΛCDM results. In the model with $w(0) < 0.9$ there is a deviation from $H_{\Lambda\rm CDM}(z)$ by at most 2.7% (marked by the horizontal line in the bottom panel) and implies a deviation in distance modulus of less than 0.05 mag. Note that the ratio $H(z)/H_{\Lambda\rm CDM}$ (bottom panel) for the double exponential potential converges to $\sim 1 + (3/8)\epsilon \sim 1.017$ due to the matter-dominated value of $\Omega_{\rm DE}(z)$ being constrained to be 3/4 of the radiation-dominated value. These illustrations show that detection will be extremely difficult even with the Stage-IV dark energy experiments such as DUNE, JDEM, LSST and SKA [66].

In contrast there is a more substantial difference between ΛCDM and positive values of μ shown by the light brown lines in Figure 4.3. Which produces a bigger deviation for $H(z)$ from $H_\Lambda(z)$ for this case of about 5%, peaking at $z \sim 1$ with a maximum value of $\Delta\mu \sim 0.9$ mag. Such a model will be detectable with Stage-III supernova surveys and with the upcoming Baryon Acoustic Oscillation (BAO) surveys such as WiggleZ, BOSS and WFMOS [84], since the maximum deviation in $H(z)$ coincides with the redshift ranges in which they will operate, i.e. $0.7 \lesssim z \lesssim 1.1$. This is only for the cases where we consider $w(0) \sim -0.8$ and if we rather look at values more consistent with the current best-fits, $w(0) < -0.9$ one finds much smaller deviations of 2.7% and 0.045 mag respectively for $H(z)$ and $\Delta\mu$ which again will require Stage-IV experiments for conclusive detection as can be seen in the bottom plot in Figure 4.4. Similar results will apply to other modifications of the exponential potential, e.g. [85].

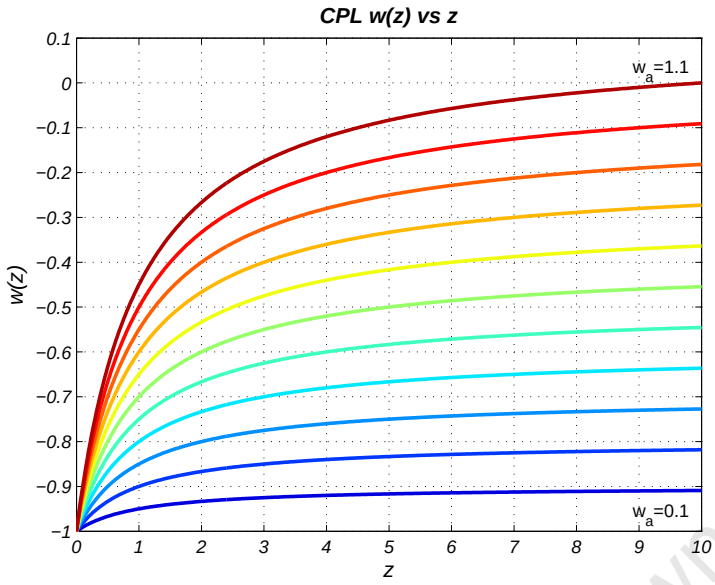


Figure 4.5: ***w(z)* for CPL parametrization** for a range of w_a values from 0.1 to 1.1 in steps of 0.1. This illustrates how the CPL parameterisation has a $w(z)$ which is convex and has a rapid transition as it approaches $w(z=0)=-1$.

4.4 CPL parameterisation

The DETF figure of merit (FoM) [66] is based on the commonly used Chevallier-Polarski-Linder (CPL) parameterisation [8, 9],

$$w(z) = w_0 + w_a \frac{z}{1+z}. \tag{4.19}$$

It would thus make sense to evaluate how this parameterisation deals with the BBN constraint combined with scaling. This however results in an unexpected and drastic failure of the CPL parameterisation to both meet the BBN constraint and demand $w(z) \geq -1$, $\Omega_{\rm DE} \sim 0.7$ today and $w(z \geq z_t) = 0$ as before (for some z_t). One can gain good intuition for this by considering Figure 4.5, here we can see that in the best instance of the CPL parameterisation going from $w_0 = 0$ to $w_0 = -1$ a rapid transition results at a low $z_t < 2$ going from $w(z) = -0.25$ to $w(z) = -1$. However if we recall the instance of the step function were the $z_t = 2.67$, this is the case of both the most rapid transition and the lowest z_t allowed by the BBN constraint and hence it becomes clear that CPL will battle to coincide with the requirements of $w_0 = -1$ and the same BBN constraint.

If we consider how this rapid transition effects CPL's ability to reach an $\Omega_{\rm DE} \sim 0.7$ today, we can quickly conclude that due to the rapid transition there isn't a long

enough period of time spent between z_t and $z = 0$ at a sufficiently negative value for $w(z)$ to force $\Omega_{\rm DE} = 0.7$. This is compounded by the fact that the shape of the $w(z)$ function for CPL is convex and thus does not favour a prolonged period at a low $w(z)$ value as would be the case for a concave function such as tanh. The inability of CPL to meet the $\Omega_{\rm DE}$ constraint is illustrated in Figure (4.3), where one can see what the initial value of $\Omega_{\rm DE}$ would have to be ~ 0.2 at z_t in order to reach $\Omega_{\rm DE} = 0.7$ today for $w_0 = -1$. We can next attempt to determine the least phantom value of w_0 that is able to satisfy the BBN constraint. Imposing the constraint $w(z_t) = 0$ and solving for w_a in terms of w_0:

$$\begin{aligned} 0 &= w_0 + \frac{w_a z_t}{1+z_t}, \\ \therefore \quad w_0 &= -\frac{w_a z_t}{1+z_t}, \\ \therefore \quad w_a &= -\left(\frac{(1+z_t)}{z_t}\right) w_0, \end{aligned} \tag{4.20}$$

Next we substitute the above result for w_a into $w(z)$:

$$\begin{aligned} w(z) &= w_0 - \left(\frac{1+z_t}{z_t}\right)\frac{w_0 z}{1+z}, \\ &= w_0\left(1 - \frac{1+z_t}{z_t}\frac{z}{1+z}\right). \end{aligned}$$

Taking the limit $z_t \to \infty$ reduces the above expression to:

$$w(z) = w_0\left(1 - \frac{z}{1+z}\right). \tag{4.21}$$

Using this result along with Eq. (4.19) we find $w_a = -w_0$. Using the standard method introduced in Section 4.2 along with Eq. (4.2) where,

$$f(z_t) = (1+z_t)^3 e^{3w_0 z_t}, \tag{4.22}$$

sets up an equation which can be solved for w_0,

$$\begin{aligned} (1+z_t)^3 e^{3w_0 z_t} &\leq \frac{\epsilon}{r(1-\epsilon)}(1+z_t)^3, \\ \Rightarrow \quad w_0 &\leq -\frac{1}{3}\ln\left(\frac{\epsilon}{r(1-\epsilon)}\right), \\ &\leq -1.30. \end{aligned} \tag{4.23}$$

Thus the highest w_0 allowed by phantom models while still matching the BBN con-

straint is $w_0 = -1.3$. In contrast the logarithmic expansion $w(z) = w_0 + w_a \ln(1+z)$ is able to match the BBN constraint with $w(z) \geq -1$, but only for $z_t > 12.4$.

4.5 A Counter Example

In the preceding sections we have shown how the BBN constraint in combination with a tracking canonical scalar field implies that a rapid low redshift transition ($z < 2.67$) is not permitted. This suggests that the prospects of detecting dynamical dark energy, assuming the above constraints, is unlikely in the next 10 years since surveys probing sufficiently deep redshifts lack the necessary sensitivity [1].

One counter example that can be considered is that of a double transition for $w(z)$ between $z = z_t$ and $z = 0$ where the second redshift transition is called z_l and will have an intermediate plateau of $w(z_l < z_t) = w_l$, see Figure 4.6. This would still adhere to the BBN constraint but could be constructed in such a way that it would allow for a rapid transition in $w(z)$ at a low redshift. This then allows for dynamical dark energy to occur at a low and thus detectable redshift. In this section we explore how viable this scenario is.

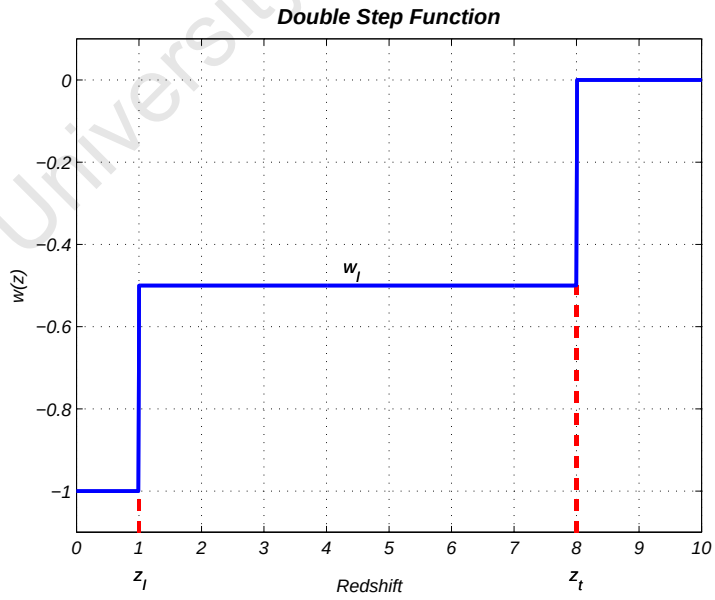


Figure 4.6: **Double Step Function** illustrating the proposed counter example $w(z)$. There are now three free parameters: z_t, z_l and w_l.

The equation we use to generate the double step function is given by,

$$w(z) = \begin{cases} 0 & z \geq z_t, \\ w_l & z_l \leq z < z_t, \\ w_0 & z \leq z_l, \end{cases} \tag{4.24}$$

where we assume $w_0 = -1$ throughout this section. Employing a similar strategy to Section 4.2 we require a relationship to relate the parameter z_t in terms of z_l, w_l and ϵ. Again the integral in $f(z) = \exp\left[3\int \frac{1+w(z')}{1+z'}dz'\right]$ is simplified by the fact that we are using a step-function, as before, albeit a double step-function in this case. Once we have solved for $f(z_t)$ we can use Eq. (4.2) to set up the following relationship:

$$\begin{aligned} \left[\frac{1+z_t}{1+z_l}\right]^{3(1+w_l)} &= \frac{\epsilon}{(1-\epsilon)}\frac{1}{r}(1+z_t)^3, \\ z_t &= \left[\frac{\epsilon}{(1-\epsilon)}\frac{1}{r}(1+z_l)^{3(1+w_l)}\right]^{\frac{1}{3w_l}} - 1. \end{aligned} \tag{4.25}$$

We can now solve Eq.(4.25) for z_t given a specified range of z_l and w_l. Since we want to test the feasibility of a double step function transition, with one being at a low redshift, we consider a range of z_l from 0.1 to 1 in steps of 0.225 and w_l from -0.5 to -0.1 in steps of 0.1. Figure 4.7 illustrates the solutions for z_t for this range of parameters. The values for z_t are so large for $z_l < 0.2$ and $w_l > -0.3$ that $\log(z_t)$ was used in the figure to better present the range of solutions.

4.5.1 CMB Constraints

Since the BBN constraints do not exclude the possibility of the double step parametrization we need another means to test its plausibility. For this we use a variant of CMBFAST [61,62], which has an additional module called KINKFAST [86]. This software allows for the evaluation of dynamical dark energy parameterisation and produces an angular power spectra which can be compared to that of the current 5 year WMAP data [11, 12]. The parameterisation used to mimic the double step function is given by,

$$w(z \leqslant z_t) = \underbrace{\left[\frac{1+w_l}{2}\right]\tanh\left(\frac{z-z_l}{\triangle}\right)}_{\text{second transition, } z_l} + \underbrace{\left[\frac{-w_l}{2}\right]\tanh\left(\frac{z-z_t}{\triangle}\right)}_{\text{first transition, } z_t} - 0.5, \tag{4.26}$$

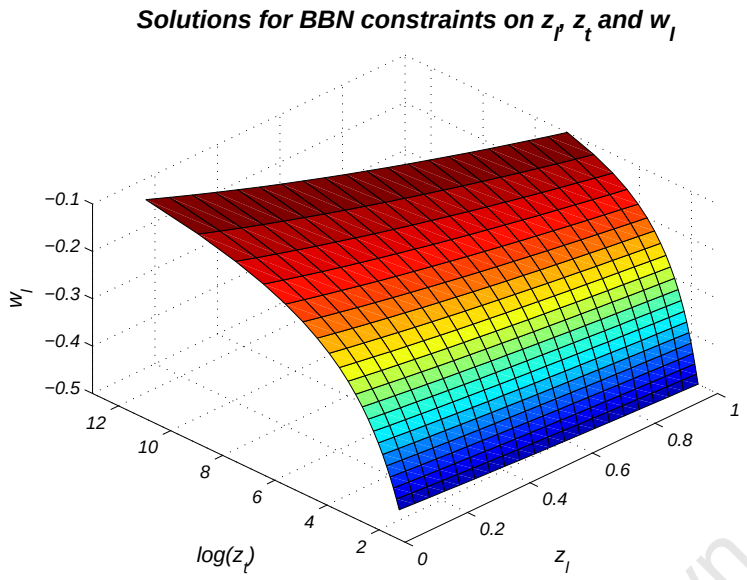


Figure 4.7: **Solutions for z_t** – This figure shows the logarithmic solution for Eq. (4.25) given the range of values for z_l from 0.1 to 1 and w_l from -0.5 to -0.1. Note how low values of $z_l < 0.2$ in combination with higher values of $w_l > -0.3$ result in large z_t values. This is so pronounced that $\log(z_t)$ is used to suppress the results sufficiently to illustrate the full range of solutions.

where Δ represents the steepness of the transition taken by each of the two steps. If the Δ is set to be too rapid ($\Delta < 0.01$) this can result in numerical instabilities or a very long run time. We initially choose a range for Δ from 0.1 to 0.3 in steps of 0.1, but this proved to have a marginal effect on the results generated and we will thus only consider the $\Delta = 0.1$ case. The appropriate values for w_l, z_l and z_t are then used in Eq. (4.26) to produce a $w(z)$ which is used to produce a corresponding angular power spectra. The range of the double step $w(z)$ parameterisation is shown in the top panel of Figure 4.8 (for $w_l = -0.1$), Figure 4.9 (for $w_l = -0.3$) and Figure 4.10 (for $w_l = -0.5$) with the corresponding power spectrum plots in the bottom panel (each COBE normalised [10]) where the same line colour is used for the matching $w(z)$. In each of these figures the blue lines represent the lowest z_l value (highest z_t value). The colours gradually move more to the red end of the spectrum as z_l increase in value until eventually reaching red in the case of $z_l = 1$ (lowest z_t value).

None of the spectra produced for the range of w_l's are even marginally within range of the WMAP angular power spectrum data and we can thus safely conclude that this type of double step parametrization is not permitted by the CMB. We can analyse each of the cases of w_l in more detail to better appreciate the reason for the

discrepancies.

There exists an interplay between w_l, z_l and z_t which is governed by the BBN constraint and is reflected in Eq. (4.25). As less negative w_l are considered in combination with small values for z_l so progressively higher values for z_t are required to solve the BBN constraint and can be seen in Figure 4.7. This is shown in the z_t's produced from the combination of w_l's and z_l's we have chosen for this counter example. The reason for z_t being so large in these instances is the same as explained in Section 4.4, which described why the CPL parameterisation failed in combination with the BBN constraint and a tracking model. Remembering that the rate at which $\Omega_{\rm DE}$ grows is governed by $w(z)$ (via the integral of $f(z)$), with a more negative $w(z)$ results in a faster growth of $\Omega_{\rm DE}$. We can thus see that in order for $\Omega_{\rm DE}$ to grow from 0.045, at transition, to 0.7 today we will require the transition to occur at progressively earlier times as w_l is made less negative ($\Omega_{\rm DE}$ grows more slowly). A large negative $w(z)$ at $z \leq z_l$ would ease the requirement for z_t to be as high but, for simplicity, we don't consider phantom models in the current counter example.

Translating the above interplay between the free parameters to the resulting lower peaks produced for the C_l's plotted is a direct consequence of the Integrated Sachs Wolfe (ISW) effect. As the dark energy dominance starts at progressively earlier times (higher zt) the gravitational potential begins to evolve at earlier times leading to a markedly increased ISW effect compared with the usual concordance model. This is shown in Figures 4.8-4.10, where the acoustic peaks at $l > 200$ are suppressed due to the strong ISW effect and the normalisation of all spectra at $l = 10$.

4.6 Conclusion

Scaling models are arguably the best-motivated alternatives to the cosmological constant. We have shown that the constraints on the energy density of a tracking scalar field at the time of Big Bang Nucleosynthesis and decoupling strongly limit their allowed dynamics. If w today is not close to the maximum value allowed by current data then detection of dynamics will likely have to wait a decade for the Stage-IV DETF experiments. Of course, these strong conclusions are only true for canonical scaling field models and if one allows exotic phantom behaviour ($w < -1$) the conclusion is much more optimistic.

We discussed two specific families of these scaling models while imposing the early Universe constraints on general scaling models is left for future work. In addition we have shown that the standard CPL parameterisation, $w(z) = w_0 + w_a z/(1+z)$, fails

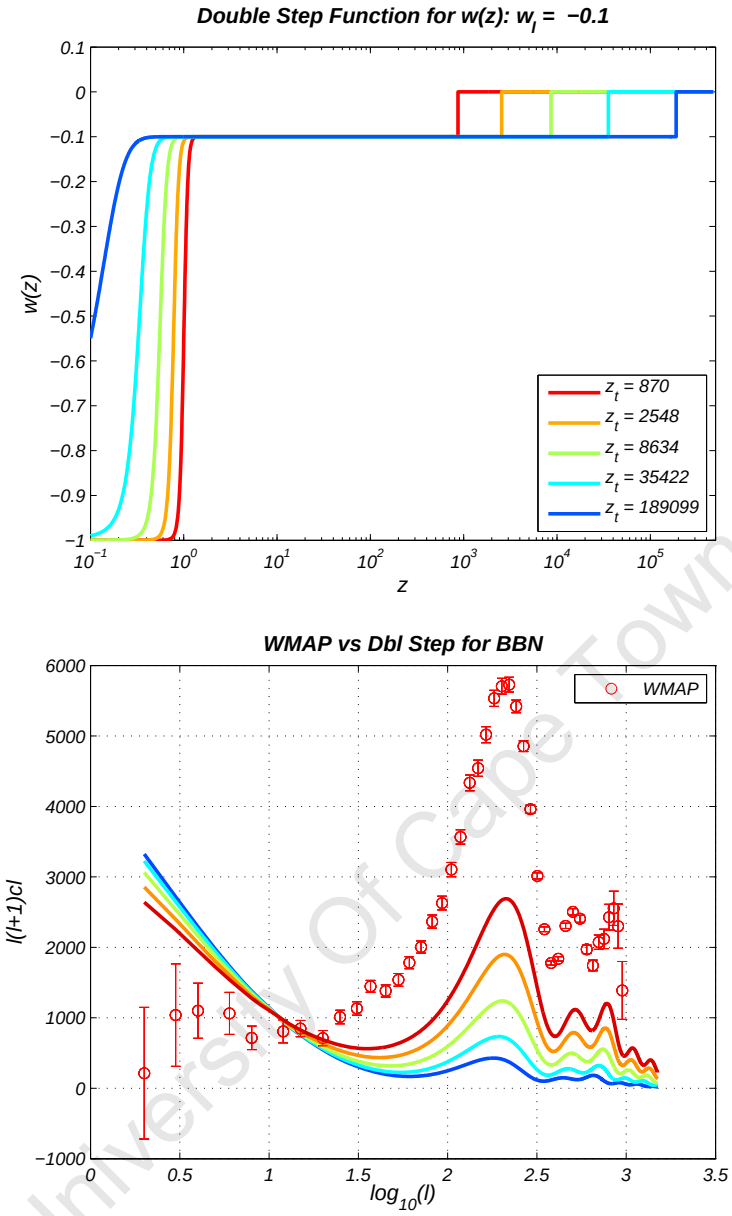


Figure 4.8: $\boldsymbol{w_l = -0.1}$ – Here we show the double step $w(z)$ parameterisation (top panel) for the range of $z_l \in [0.1, 1.0]$ (in steps of 0.225) for a $\Delta = 0.1$ producing a range of z_t's, shown in the legend, calculated from Eq. (4.25). A logarithmic scale for z is used in the top panel due to the broad range of z_t's produced. This creates the illusion of sharper transition at high z_t values as opposed to the low z_l values but the transition for both are equally steep as illustrated in Figure 4.6 where a linear scale is used. The colours represent the respective value of z_t from blue for the highest to red for the lowest value. The associated colour for each z_t is used to plot the C_l (bottom panel) which are each COBE normalised [10] and compared to the WMAP 5 year data [11, 12]. Here we can see that the highest z_t (blue line) is furthest from matching the WMAP data (red data markers) while as the value for z_t decreases (become more red) so the peaks progressively move closer to, but don't match, the corresponding WMAP data. This is a consequence of the Integrated Sachs Wolfe (ISW) effect, introduced in Section 2.3.

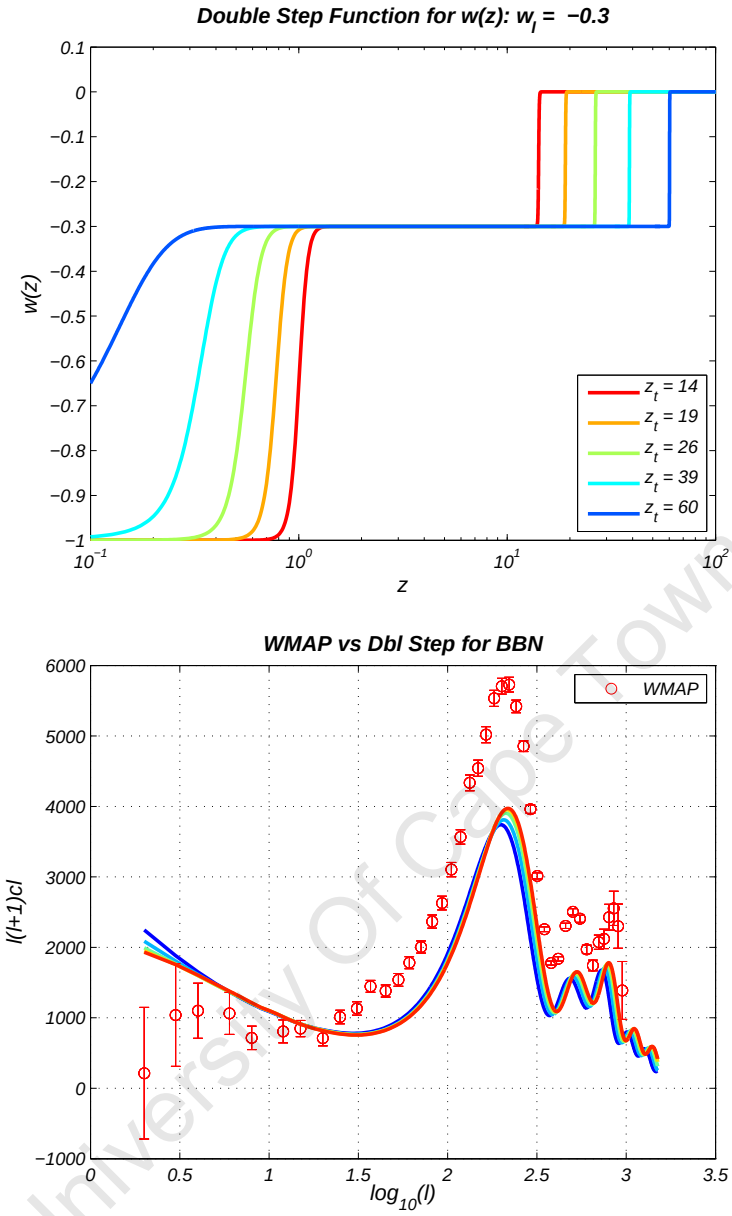


Figure 4.9: $\boldsymbol{w_l = -0.3}$ – Here we show the double step $w(z)$ parameterisation (top panel) for the range of $z_l \in [0.1, 1.0]$ (in steps of 0.225) for a $\Delta = 0.1$ producing a range of z_t's shown in the legend produced from Eq. (4.25). A logarithmic scale for z is used in the top panel due to the broad range of z_t's produced. This creates the illusion of sharper transition at high z_t values as opposed to the low z_l values but the transition for both are equally steep as illustrated in Figure 4.6 where a linear scale is used. The colours represent the respective value of z_t from blue for the highest to red for the lowest value. The associated colour for each z_t is used to plot the C_l (bottom panel) which are each COBE normalised [10] and compared to the WMAP 5 year data [11, 12]. Again the highest value of z_t (blue line) is furthest from matching the C_l's of the WMAP data while the lowest value of z_t is the closest again via the ISW effect. There is a drastic narrowing in the range between the peaks of the differing z_t's compared to Fig. 4.8. This is due to a smaller difference in the z_t's produced in the instance of $w_l = -0.3$.

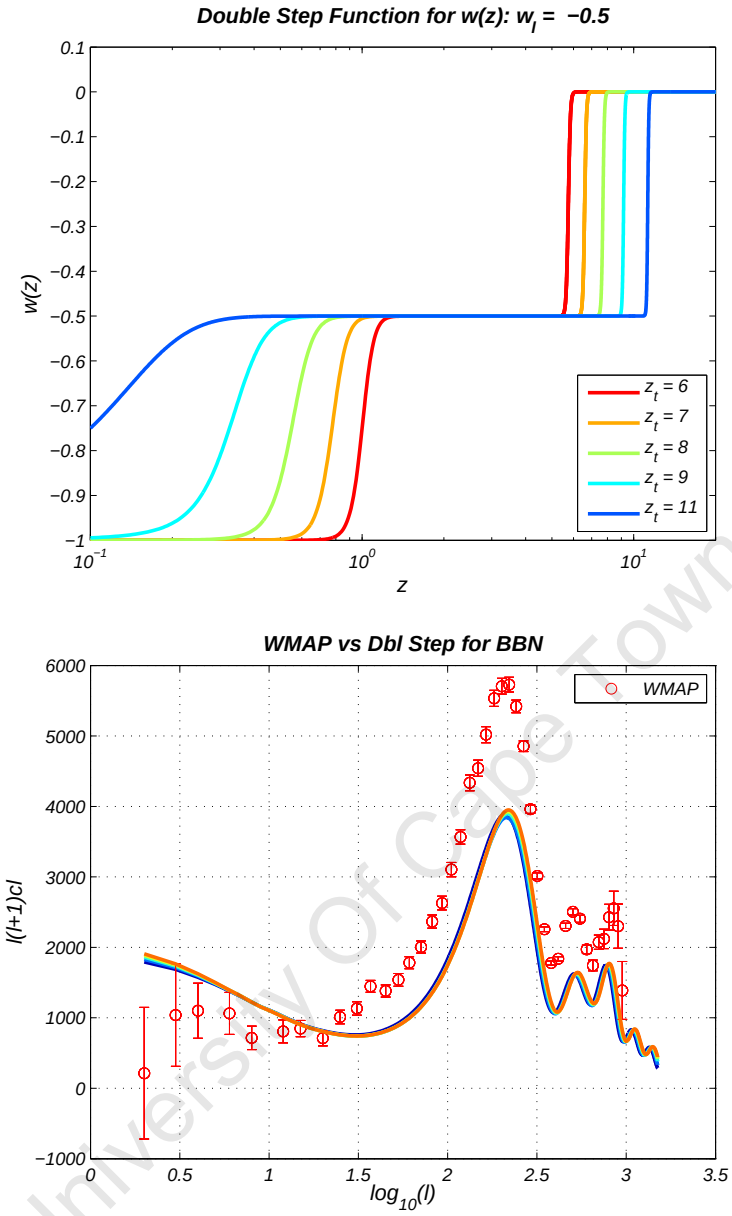


Figure 4.10: $\boldsymbol{w_l = -0.5}$ – Here we show the double step $w(z)$ parameterisation (top panel) for the range of $z_l \in [0.1, 1.0]$ (in steps of 0.225) for a $\Delta = 0.1$ producing a range of z_t's shown in the legend produced from Eq. (4.25). A logarithmic scale for z is used in the top panel due to the broad range of z_t's produced. This creates the illusion of sharper transition at high z_t values as opposed to the low z_l values but the transition for both are equally steep as illustrated in Figure 4.6 where a linear scale is used. The colours represent the respective value of z_t from blue for the highest to red for the lowest value. The associated colour for each z_t is used to plot the C_l (bottom panel) which are each COBE normalised [10] and compared to the WMAP 5 year data [11, 12]. This produces the same predictable trend as described in the previous two figures with the highest value of z_t (blue line) being the furthest from matching the C_l's of the WMAP while the lowest value of z_t is the closest as a result of the ISW effect. There is however the most notable narrowing in the range between the peaks of the differing z_t's in this example of $w_l = -0.5$. This is again attributed to the much smaller spread of different z_t's produced.

dramatically to match the BBN constraint when describing scaling fields which satisfy $w \geq -1$. This is particularly important given that the CPL parameterisation is the basis of the DETF figure of merit [66] which is now the *de facto* standard for the optimisation of future cosmological surveys, e.g. [87]. A concern therefore is that optimisations may be unwittingly biased *away* from scaling dark energy models.

The double step function parameterisation of $w(z)$ is introduced as a possible counter example to allow for a rapid transition of w at a lower and thus more detectable redshift range. Although a solution of this nature does adhere to the BBN constraints it was found that it cannot match the CMB constraints and fails to produce a power spectrum that can agree to the WMAP data [11, 12]

More work in this area is clearly needed to assess the implications for cosmological survey design, but it is no surprise that the current non-detection of dark energy dynamics should neither come as a surprise, nor should it discourage us from the hunt.

Chapter 5

Fisher Matrix Cosmology using Fisher4Cast

This chapter is composed of work presented in [2] with a supplemented explanation of the nature of the rotation of the ellipses described in Section 5.3.

Fisher Matrix theory is a powerful tool introduced in Section 1.1.8 along with some foundational theory and notation which we follow for the remainder of this thesis. Fisher4Cast is a software suite designed to help do general Fisher Matrix calculations that also lends insight for students and researchers alike, whether for quick and easy evaluation or more advanced applications and survey forecasting. Fisher4Cast is written in Matlab[1] following an object-oriented model and using standard software engineering standards for implementation and testing. The code is not specific to cosmology; Fisher matrices can be generated given any parameter $\mathbf{X}(\boldsymbol{\theta})$. The Fisher derivatives $\partial\mathbf{X}/\partial\theta_{\mathrm{A}}$ are computed analytically (if they are known) or numerically, for example in the case of growth, allowing the code to handle complex cases without analytical formulae for $\mathbf{X}$. The code suite includes a Graphical User Interface (GUI), which is specific to the cosmological example which we discuss in the following Section 5.1. The use, design and structure of the program are detailed in the following chapter.

Fisher4Cast facilitates novel research and education in two different ways. Apart from being well-tested against existing Fisher Matrix results, it is general and modular. Because of this modular nature, a natural application of Fisher4Cast is to visualisation. The code can easily be called repeatedly in large loops, enabling one to study large-scale properties of the Fisher Matrix for a wide range of surveys and cosmologies. We give examples of such studies in the following subsection. Secondly,

[1] See http://www.mathworks.com.

Fisher4Cast is coded for a general Friedmann-Lemaître-Robertson-Walker (FLRW) universe since the curved case has rarely been studied in the literature. We study the issues of curvature in dark energy Fisher analysis in the last subsection.

5.1 The Cosmology of Hubble, Distance and Growth

Although Fisher4Cast is a completely general Fisher Matrix framework at the command-line level, the GUI is coded as a cosmology interface, since this is its primary application. In the context of modern cosmological surveys, the primary observables are the expansion rate of the Universe, measured through the Hubble rate $H(z)$, cosmological distances such as the angular diameter distance, $d_A(z)$, and the growing mode of dark matter density perturbations, $\delta(\mathbf{x}, z) \propto G(z)$. $H(z)$ and $d_A(z)$ are provided by Baryon Acoustic Oscillations (BAO) surveys while growth can be measured using lensing or number count surveys and potentially also BAO if the bias is measured independently (e.g. through redshift distortions - see [88, 89]).

The Fisher4Cast GUI uses the observables H, d_A and G in a general FLRW universe. The cosmic parameters assumed for the GUI are $(H_0, \Omega_m, \Omega_k, w_0, w_a)$, where H_0 is the value of the Hubble constant in $\mathrm{kms}^{-1}\mathrm{Mpc}^{-1}$, Ω_m is the energy density of matter today in units of the critical density, Ω_k is the curvature energy density $(\Omega_{\mathrm{DE}} = 1 - \Omega_m - \Omega_k)$[2] and w_0, w_a are the coefficients in the Chevalier-Polarski-Linder (CPL) expansion of the dark energy equation of state [8, 9]:

$$w(z) = w_0 + w_a \frac{z}{1+z} = w_0 + w_a \left(1 - \frac{a}{a_0}\right) , \tag{5.1}$$

where $a_0 = c/(H_0\sqrt{|\Omega_k|})$ is the curvature radius of the cosmos. Other dark energy expansions can be easily accommodated in Fisher4Cast by changing the appropriate input functions. The expansion history of a FLRW universe is described by the Hubble parameter:

$$H^2(z) = H_0^2 E^2(z) \equiv H_0^2 \left(\Omega_m (1+z)^3 + \Omega_k (1+z)^2 + (1 - \Omega_m - \Omega_k) f(z, w_0, w_a)\right) , \tag{5.2}$$

[2] Unless explicitly indicated elsewhere, references in this chapter to Ω_i will mean the current value of the density parameter and not its value as a function of time.

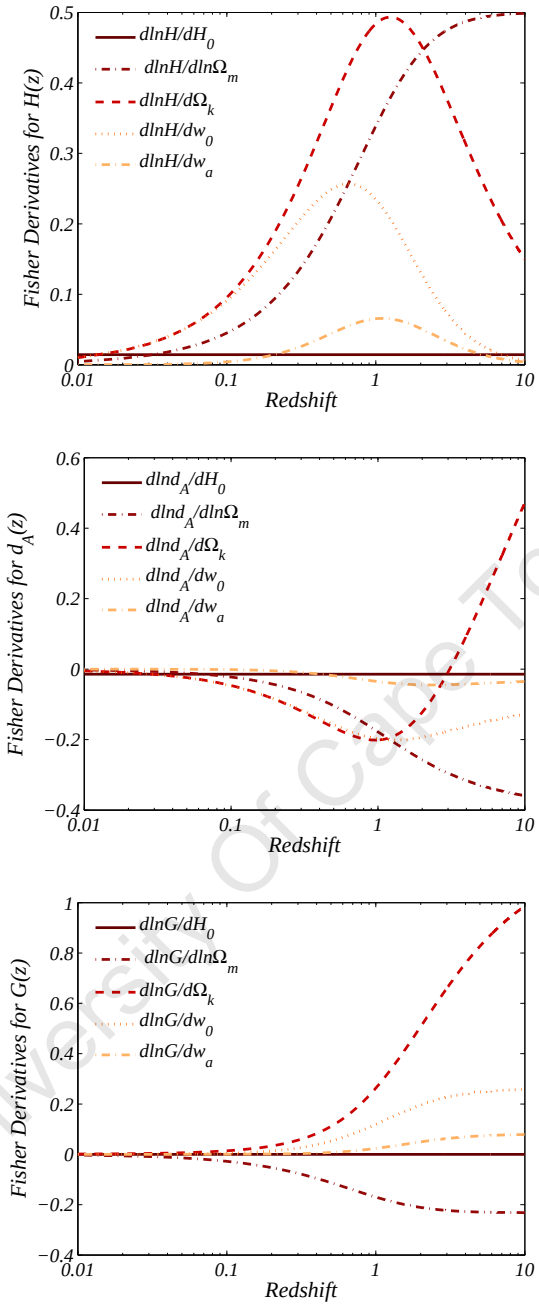


Figure 5.1: **Fisher Derivatives [2]** – The Hubble parameter (top panel), angular diameter distance (middle panel) and growth function (bottom panel) are shown for the parameters considered in our cosmological example: H_0 (solid dark red line), $\ln \Omega_m$ (dot-dashed red line), Ω_k (dashed dark orange line), w_0 (dotted orange line) and w_a (dot-dashed peach line). The full set of analytical derivatives of the $H(z)$ and $d_A(z)$ are found in Appendix C. In the case of the growth function $G(z)$ the derivatives are calculated numerically - see [2] for details.

with the evolution of the dark energy density, $\rho_{\rm DE}(z) \propto f(z)$ determined by,

$$f(z) = \exp\left(3\int_0^z \frac{1+w(z')}{1+z'}dz'\right). \tag{5.3}$$

For the CPL parameterisation, Eq. (5.1), $f(z)$ is given by,

$$f(z) = (1+z)^{3(1+w_0+w_a)} \exp\left\{-3w_a\frac{z}{1+z}\right\}. \tag{5.4}$$

The angular-diameter distance, $d_A(z)$ relates the angular size of an object to its known length, providing a measure of the distance to the object, and is given by:

$$d_A(z) = \frac{1}{1+z}\frac{c}{H_0\sqrt{\Omega_k}}\sinh\left(\sqrt{\Omega_k}\chi(z)\right), \tag{5.5}$$

where

$$\chi(z) \equiv \int_0^z \frac{dz'}{E(z')}, \tag{5.6}$$

and $E(z)$ is as defined in Eq. (5.2). These forms are valid for all values of Ω_k via continuity and the trigonometric identity $\sinh(ix) = i\sin(x)$. The often-used equation for the angular diameter distance contains three equations, depending on the sign and magnitude of Ω_k, however this is redundant, at least conceptually. In numerical analysis we use the Taylor series expansion for very small Ω_k from [2],

$$\left.\frac{\partial d_A(z)}{\partial\Omega_k}\right|_{\Omega_k\to 0} = \frac{c}{H_0}\frac{1}{1+z}\left\{\frac{1}{6}\chi^3(z,0) + \frac{\partial\chi(z,0)}{\partial\Omega_k}\right\}, \tag{5.7}$$

where,

$$X(z,0) \equiv X(z)|_{\Omega_k\to 0}, \tag{5.8}$$

are the functions (for example $E(z),\ \chi(z)$) assuming flatness.

Finally we discuss the governing equation for the growth of structure, a potentially powerful probe of dark energy [66,90–99]. In general, one needs to solve the differential equation for the perturbations in the matter density δ (assuming the pressure and pressure perturbations of the matter are zero i.e. $p = \delta p = 0$) [100–102]:

$$\ddot{\delta} + 2H\dot{\delta} = 4\pi G\rho_m\delta. \tag{5.9}$$

We discuss this equation in the context of a curved universe with dynamical dark energy in [2]. Fisher4Cast takes as input constraints on the growth $G(z)$ which provides the temporal evolution of density perturbations, i.e. $\delta(\mathbf{x}, z) \propto G(z)$.

5.2 Future Survey Constraints

Fisher4Cast can be used to help quickly generate a comparison between planned future surveys and even optimize the design specifications. This is made easier with the addition of a new extension for Fisher4Cast which is designed for calculating errors on H and d_A from BAO survey specifications.

As an example we show a comparison of the WFMOS, BOSS and WiggleZ surveys [15, 16, 103] with the error ellipses shown in Figure 5.3 using the survey specifications listed in Table 5.1-5.3. From this figure we can quickly then analyse the expected constraints that each of the respective surveys can produce for w_0 and w_a. The results show the WiggleZ survey produces a bigger error ellipse (DETF FoM ~ 0.17) than that of BOSS (DETF FoM ~ 1.86) and WFMOS (DETF FoM ~ 1.68) which are similar in size. In addition to comparing future surveys, error ellipses for all the cosmic parameters can be shown for a single survey, as illustrated in Figure 5.2. In this case the fisher ellipses have been computed for the flat-ΛCDM fiducial model - $(H_0, \Omega_m, \Omega_k, w_0, w_a) = (70, 0.3, 0, -1, 0)$ - and data as listed in Table 5.1 while using the full set of analytical derivatives for $H(z), d_A(z)$ as given in Appendix B.1.

Parameter	**Value**
Redshifts of measurement	$H : z = [0.3, 0.6, 0.8, 1.0, 1.2, 3]$ $d_A : z = [0.3, 0.6, 0.8, 1.0, 1.2, 3, 1000]$
Percentage error	**Value** [%]
$H(z)$	$\sigma_H/H = [5.80, 5.19, 3.59, 2.84, 2.53, 1.48]$
$d_A(z)$	$\sigma_{d_A}/d_A = [5.19, 4.30, 3.22, 2.3, 2.03, 1.19, 0.22]$
Cosmological model	**Value**
$(H_0, \Omega_m, \Omega_k, w_0, w_a)$	$(70\text{kms}^{-1}\text{Mpc}^{-1}, 0.3, 0, -1, 0)$
Priors on model	$(1, 100, 10^4, 0, 0)$

Table 5.1: **WFMOS-like survey data from the Seo & Eisenstein survey configuration [14]** – used in Figures 5.2, 5.3, 5.12 and 5.7. In some cases measurements of the growth function were added, taken at the same redshifts as the Hubble parameter; in others the prior information on various parameters was changed. See the captions of the relevant figures for the specific details. The priors, as explained in Section 1.1.8, are directly input into Fisher4Cast as a diagonal matrix and have been chosen to be more conservative than the original data.

5.3 Rotation of Ellipses

Understanding the role the Fisher Matrix plays in the orientation, width and length of the error ellipse becomes important in the following section. The eigenvectors and

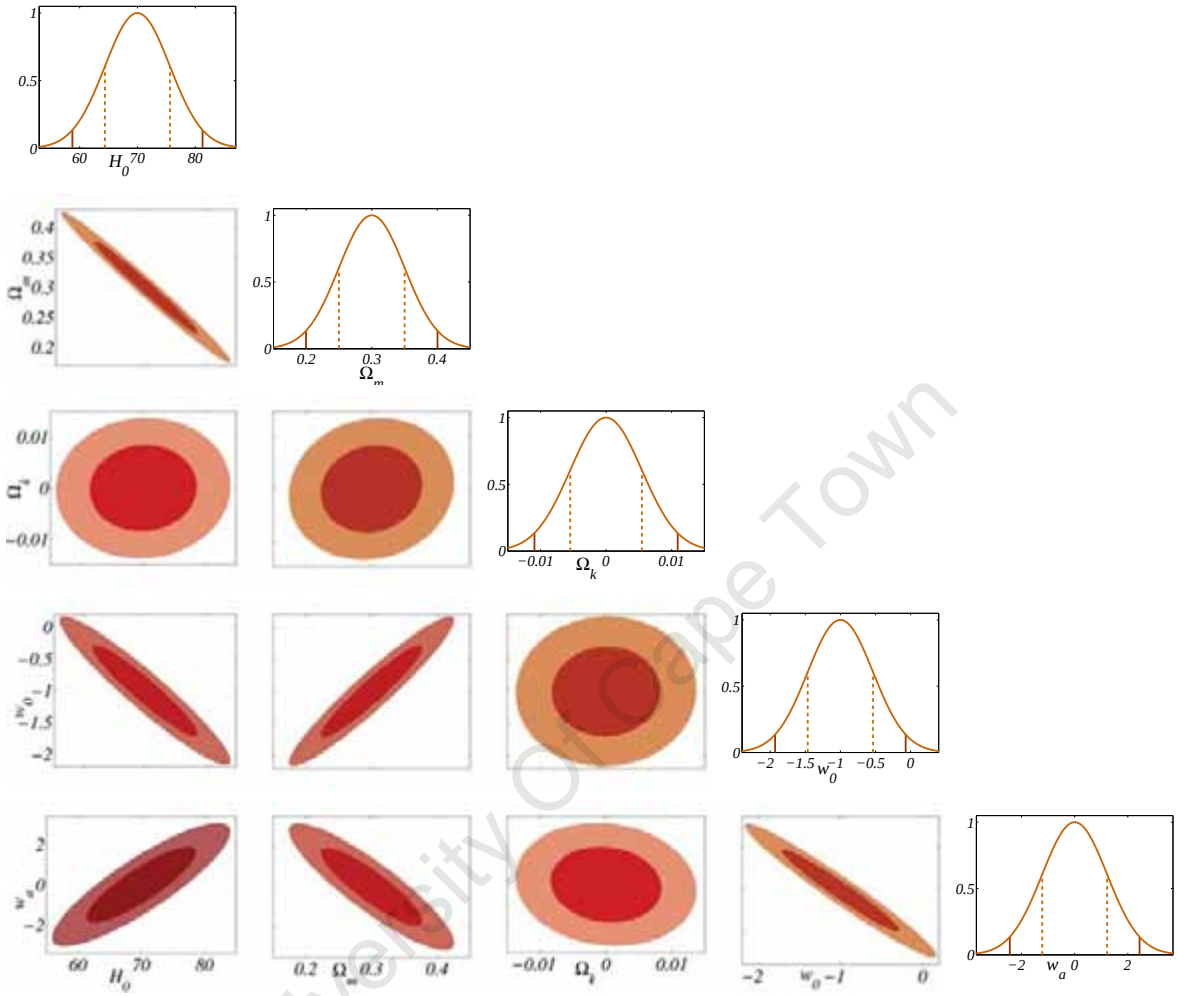


Figure 5.2: **Marginalised Fisher ellipses for all parameters [2]** – Here the full range of Fisher ellipses are produced by Fisher4Cast for the cosmological parameters, $H_0, \Omega_m, \Omega_k, w_0, w_a$, using data from Table 5.1 and including growth measurements with redshifts and errors the same as given for H_0. The $1-$ and $2-\sigma$ confidence levels are respectively represented by dark inner and light outer shaded contours and are marginalised over all other parameters. The fully marginalised one-dimensional likelihood for each parameter is shown along the diagonal, where the 1 and $2-\sigma$ limits are denoted by solid and dashed vertical lines respectively.

Parameter	**Value**
Redshifts of measurement	$H : z = [0.35, 0.6, 2.5]$ $d_A : z = [0.35, 0.6, 2.5]$
Percentage error	**Value** [%]
$H(z)$	$\sigma_H/H = [1.0, 1.1, 1.5]$
$d_A(z)$	$\sigma_{d_A}/d_A = [1.8, 1.7, 1.5]$
Cosmological model	**Value**
$(H_0, \Omega_m, \Omega_k, w_0, w_a)$	$(70\text{kms}^{-1}\text{Mpc}^{-1}, 0.3, 0, -1, 0)$
Priors on model	$(1, 100, 10^4, 0, 0)$

Table 5.2: **BOSS-like survey data [15]** – used in Figure 5.3 to depict the blue inner ellipse. The priors listed here are input into Fisher4Cast as a diagonal matrix where higher numbers represent a great amount of prior information for an associated parameter value of the model (see Section 1.1.8 for more details).

Parameter	**Value**
Redshift of measurement	$H : z = [0.7]$ $d_A : z = [0.7]$
Percentage error	**Value** [%]
$H(z)$	$\sigma_H/H = [1.8]$
$d_A(z)$	$\sigma_{d_A}/d_A = [2.7]$
Cosmological model	**Value**
$(H_0, \Omega_m, \Omega_k, w_0, w_a)$	$(70\text{kms}^{-1}\text{Mpc}^{-1}, 0.3, 0, -1, 0)$
Priors on model	$(1, 100, 10^4, 0, 0)$

Table 5.3: **WiggleZ-like survey data [16]** – used in Figure 5.3 to depict the larger outer ellipse. The priors listed here are input into Fisher4Cast as a diagonal matrix where higher numbers represent a great amount of prior information for an associated parameter value of the model (see Section 1.1.8 for more details).

eigenvalues of the Marginalised Fisher Matrix (MFM) are the determining factors in the orientation and dimensions of the error ellipses. This forms an important aspect of the analysis in the following section and thus the mechanics of how this occurs is addressed in this section. The relationship of the eigenvectors and eigenvalues with that of the ellipse is summarised in Figure 5.4. The eigenvectors, γ_1 and γ_2, of the MFM orientate the semi-minor and semi-major axes while the eigenvalues, λ_1 and λ_2, contribute to the length of the semi-minor and semi-major axes respectively. Here d^2 in the figure is equivalent to the confidence level under consideration, e.g. $d^2 = 2.30$ for a 1σ confidence level and $d^2 = 6.17$ for a 2σ confidence level.

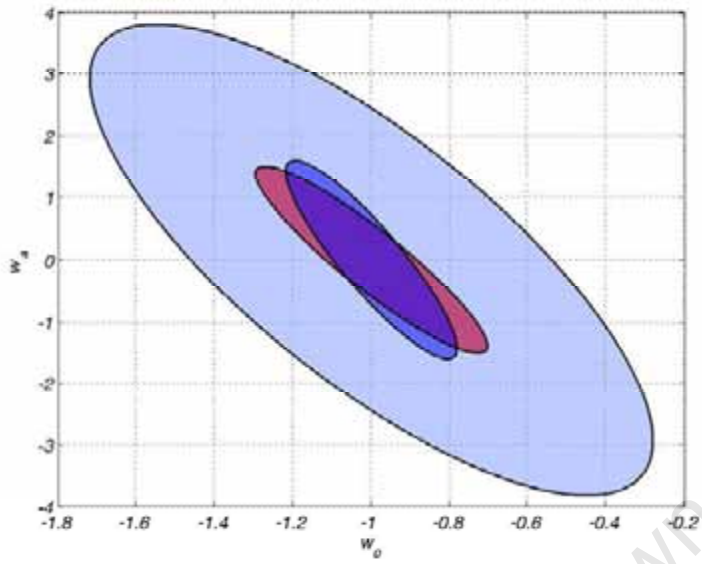


Figure 5.3: **Future Survey Comparison** – Here the WiggleZ, BOSS and WFMOS surveys are compared to show the different error ellipses produced for w_0 and w_a. The larger ellipse corresponds to the WiggleZ survey while the smaller red ellipse is for the WFMOS survey and lastly the inner blue ellipse is the associated ellipse for the BOSS survey.

Consider an example where we only have a single redshift bin for H or d_A and vary this bin value from $z = 0.1$ to $z = 5.0$. This does not result in a closed error ellipse and produces parallel lines indicating the degeneracy direction of the observables. This is a good first example which lays a foundation for the more complex example presented in the next section. To get a graphical appreciation of the rotation of the ellipse changing as the redshift is varied, see Figure 5.5. The low redshift results are plotted in blue and progressively becomes more red as the redshift increases. We note a rapid anti-clockwise rotation for H at low redshift while the rotation of d_A, although in the same direction, is more subdued and has little rotation evident at higher redshifts.

The reason for the rotation seen in Figure 5.5 is due to the MFM generated which is different at each redshift and thus results in differing eigenvalues and eigenvectors at each z. This in turn impacts the rotation and width of the error ellipses.

We can analyse the change in eigenvalues and eigenvectors for the MFM as a function of redshift to more systematically illustrate the nature of the rotation of the ellipses and change in their width for a varying redshift bins. This is done in Figure 5.6 for both the observables H (red lines) and d_A (black dashed lines).

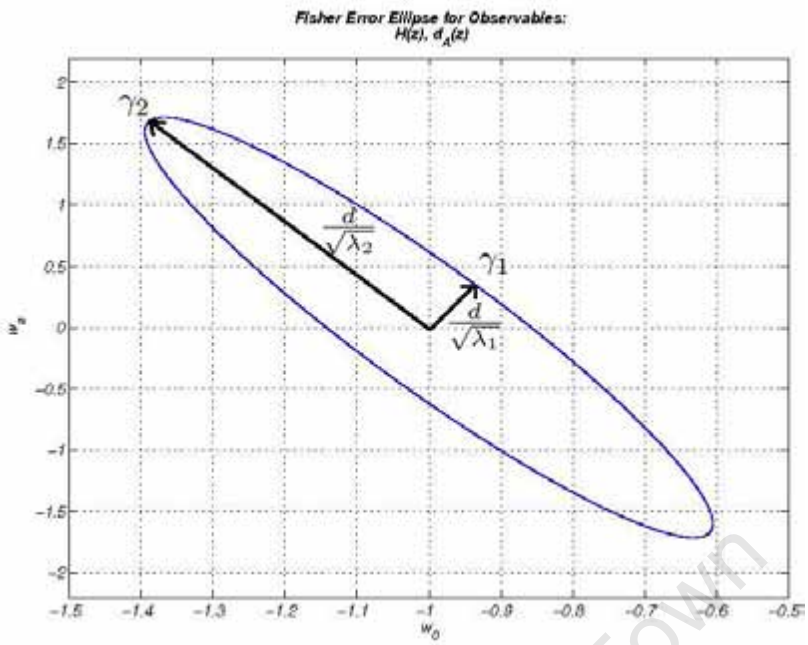


Figure 5.4: **Eigenvalues and eigenvectors for an ellipse** – This figure shows the eigenvectors γ_1 and γ_2 orientating the alignment of the ellipse. The eigenvalues λ_1 and λ_2 are responsible for determining the width of the ellipse along the semi-minor and semi-major axes respectively. The value for d is dependent on the confidence level chosen for σ e.g. $d^2 = 1\sigma = 2.30$.

Figure 5.6 shows how the angle of the γ_2 eigenvector deviates from the vertical axis in degrees over a range of redshifts from $z = 0.1$ to $z = 5.0$, in the left panel. The orientation of the semi-major axis is shown to change, in the same way as depicted in Figure 5.5, but quantifies in degrees how the angle changes from the vertical axis. The region with the steepest gradient corresponds to the redshift range in which the fastest rotation takes place. This mirrors the results shown Figure 5.5 where a rapid low redshift rotation was observed for H and almost no rotation for high redshift of d_A.

In the right panel of Figure 5.6 the eigenvalues for H and d_A are plotted as a function of redshift. The peaks of 0.75 and 1.25 for H and d_A represent the redshift at which the degenerate lines (semi-minor axis of the ellipse) will be at their narrowest. As previously described, a single redshift bin for the observables H and d_A doesn't produce a closed error ellipse but rather results in two parallel lines. This is reflected in the eigenvalues for λ_2 which are zero for all redshifts and thus not plotted in Figure 5.4. This produces a semi-major axis which is infinite in length for the ellipse and thus implies that we have parallel lines. The length of the semi-minor axis is however well defined and will be at its narrowest at the peak of λ_1. This peak exists

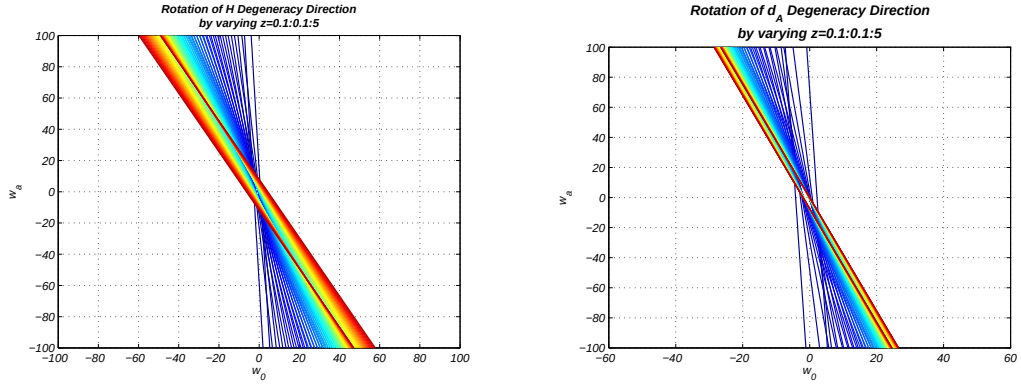


Figure 5.5: **Varying redshift of H and d_A** – Here the degeneracy direction is shown as a function of redshift for both observables H (left panel) and d_A (right panel) where only a single redshift bin is considered. The colder colours, starting with blue, represent the orientation at the lowest redshift bin of $z = 0.1$ and progress to the hotter colours ending with red at $z = 5.0$. A fast rotation is apparent for the degeneracy direction of H at lower redshifts while it then becomes more subdued at higher redshifts. In contrast the rotation of d_A is far less pronounced across the redshift range from $z = 0.1$ to $z = 5.0$.

at a redshift of about $z \sim 0.75$.

5.4 Visualisations

One of the key design principles behind Fisher4Cast is ease of use where the GUI was specifically created to provide users without an in-depth knowledge of Fisher Matrix theory access to the power of the formalism. Fisher4Cast can also be called from the command-line, as shown in the code examples in Section 6.1.1 of the following chapter. In this subsection we investigate applications of Fisher4Cast to probe and visualise the Fisher Matrix. Figure 5.7 illustrates the resulting ellipses when Fisher4Cast is called in a simple two-dimensional loop that varies the values of $w_0 - w_a$ in the cosmological model assumed to be true. The strong dependence of both the orientation and size of the resulting ellipse on the assumed model is clearly evident, as the ellipses rotate and shrink for larger values of w_0 and w_a.

Figure 5.8 illustrates the complementary case when the assumed cosmological model is kept fixed, but where instead the survey parameters are varied. In all cases the survey consisted of one measurement of $d_A(z)$ and $H(z)$ in a single redshift bin (with fractional errors of 10% on either observable), while the redshifts of these two bins were varied independently over the range $0.1 < z < 5$. The DETF Figure of Merit (FoM) of the survey ($\propto$ 1/Area of the $w_0 - w_a$ ellipse) is plotted as a FoM

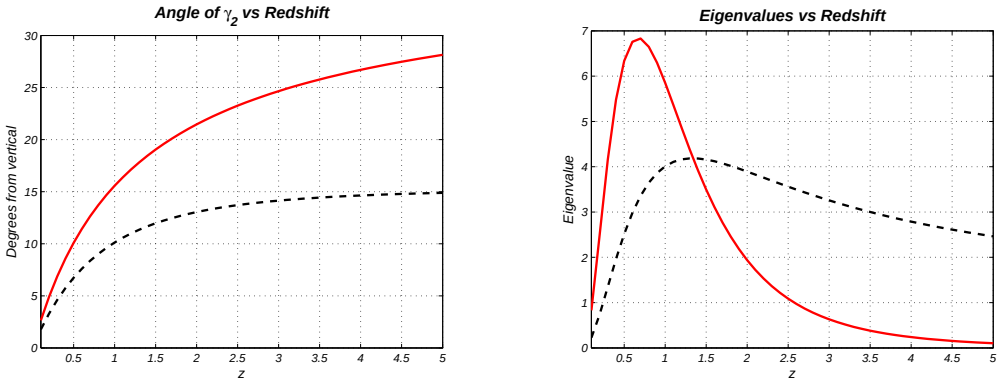


Figure 5.6: **Angle of γ_2 and eigenvalue λ_1 for H and d_A** – The angle of the eigenvector γ_2 from the vertical axis is plotted in the left panel as a function of the redshift bin for both H (red solid line) and d_A (black dashed line) changing from $z = 0.1$ to $z = 5.0$. The steep gradient in the range of $z = 0.1 - 1.5$ for H corresponds to the rapid rotation observed in Figure 5.5 (left panel) at low redshift (blue shades). The eigenvalues in the right panel show a peak for λ_1 at $z \sim 0.75$ for H and $z \sim 1.25$ for d_A which indicated the redshift at which the semi-minor axis will be at its narrowest. Since the ellipse is not closed for the case of a single redshift bin for H or d_A, only parallel lines are produced showing the degeneracy direction. This is as a results of λ_2 being zero at all redshift (not shown).

"landscape", where the colourmap is related to the value of the FoM, where the higher values of the FoM are depicted in red, and the lower values in blue. The FoM ranges between 10^{-5} and ~ 0.08. The right-hand panel of Figure 5.8 shows this landscape projected into two dimensions, with the same FoM colourmap as in the three dimensional case.

Two interesting features are immediately apparent. First, the peak in the FoM landscape, or "FoM hotspot" occurs for a survey with measurements of $H(z)$ at $z = 1.5$ and $d_A(z)$ at $z = 0.6$, while a ridge of moderately good values of the FoM (relative to the average) emerges along the line corresponding to $H(z = 0.2)$ for measurements of d_A and redshifts larger than 1.6. Secondly these two regions of higher Figure of Merit are separated by a "cold valley" of lower FoM values.

This is of particular interest given the optimisation of current and future Baryon Acoustic Oscillation (BAO) surveys [24, 66, 104–106]. Future BAO surveys will measure the radial and tangential oscillation scale (see [107] for a recent review of BAO in cosmology), producing measurements of d_A and H at the same redshifts, i.e. along the diagonal line in this plot. For z roughly between 0.5 and 3, this line intersects regions of relatively high FoM. This landscape is explained by the geometric interplay between the size and orientation of the degeneracy directions between the $H(z)$ and

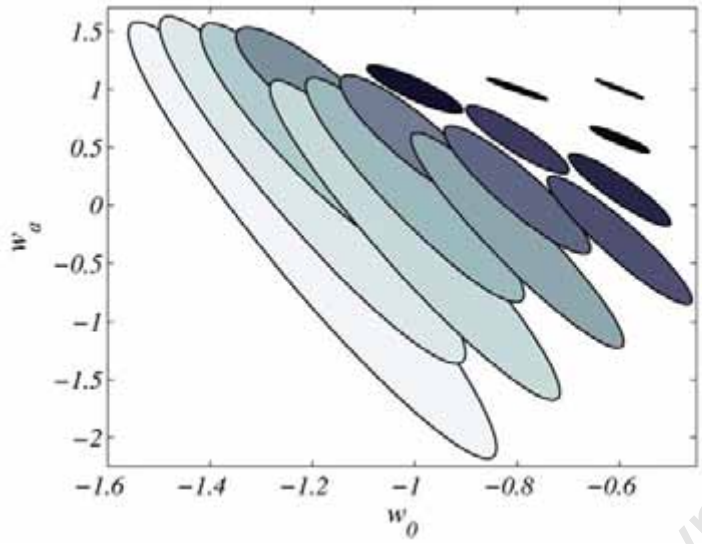


Figure 5.7: **Varying w_0 and w_a of the fiducial model** – the generated ellipses for a measurements of the Hubble parameter $H(z)$ and angular diameter distance $d_A(z)$ survey characterised in Table 5.1. The values for the coefficients in the CPL parameterisation for $w(z)$, w_0 and w_a, have been varied on a grid over $-1.3 < w_0 < -0.6, -0.7 < w_a < 1$. As w_0, w_a change, not only does the ellipse centre shift, but the size of the ellipse changes, as well as the slope of the degeneracy direction between the two parameters. Since the Dark Energy Task Force Figure of Merit is linked to the inverse of the area of the ellipse the value of the FoM increases as w_0 and w_a increase.

$d_A(z)$ ellipses, as has been discussed in the previous section and further highlighted in Figures 5.10 and 5.11.

Finally we extend the $H - d_A$ landscape to include a loop over the redshift of a single growth measurement in the range $0.1 < z < 5$. In all cases we normalise the growth today; $G(z = 0) = 1$. The full four dimensional surface cannot be plotted in general, we show slices through the hypersurface in Figure 5.9, illustrating how a measurement of the growth at high-redshift leads to larger values of the FoM overall, and opens up interesting new "hotspots". These are only a few of the potential visualisation applications for Fisher4Cast.

The FoM landscape in Figure 5.8, and in particular the valleys separating high points in FoM space warrant further investigation. We argued previously that the variation in the FoM was an interplay between the widths of the ellipses (the ellipses for a single redshift measurement are infinitely degenerate along the semi-major axis) and the orientation of the H and d_A ellipses.

This is illustrated in Figure 5.10 where we show the two degenerate ellipses (red

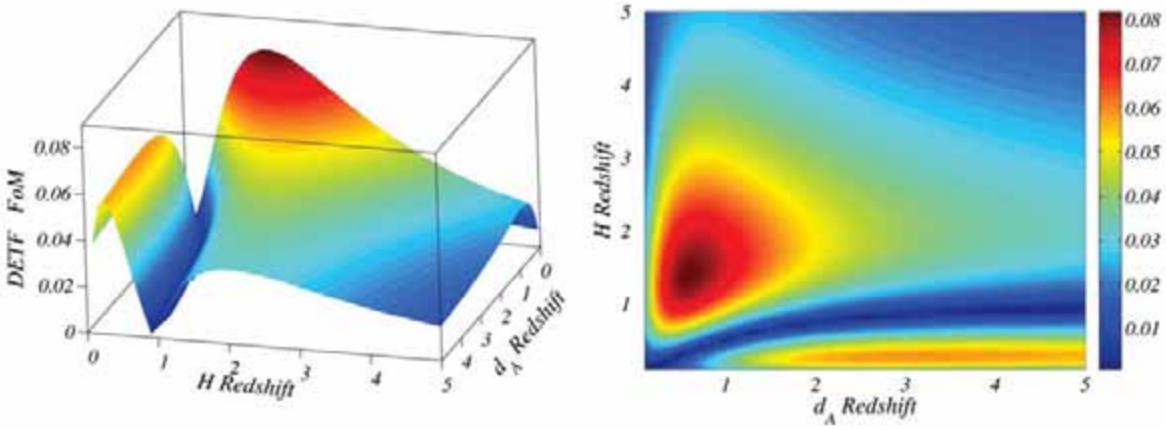


Figure 5.8: **Figure of Merit plane** – the Dark Energy Task Force Figure of Merit (FoM) for a survey consisting of one bin each of the Hubble parameter and angular diameter distance, with the fractional errors $\sigma_H/H = \sigma_{d_A}/d_A = 0.1$. The redshifts of the H, d_A measurements are varied separately from $z = 0.1$ to $z = 5$, and the resulting FoM for each survey configuration is plotted as a 3-dimensional landscape in the left panel, or a flat 2-dimensional plane in the right panel. The colourmap in both panels reflects the value of the FoM, from low values of FoM ~ 0(blue) to higher values (FoM ~ 0.08) .

solid lines for H and black dashed lines for d_A) and the combined ellipse for a series where we keep the d_A bin fixed at $z = 0.6$ while varying H in the redshift range $0.1 < z < 5$. This moves us along a line in the landscape that intersects the peak at an H bin of $z = 1.5$. The combined ellipses have colours that correspond to the FoM in Figure 5.8, the highest value of the DETF FoM (~ 0.08) coloured in red, and the lowest (~ 0) coloured in dark blue. One can see a distinct rotation of the degeneracy direction for H as the redshift changes. This anti-clockwise direction rotation is most pronounced at low redshift from $z = 0.1$ to $z = 1.0$ after which it becomes more subtle and slows down drastically at higher redshifts. This is as expected from our analysis in Section 5.3 and depicted in Figure 5.6.

Similarly in Figure. 5.11 we again show two degenerate ellipses (red solid lines for H and black dashed lines for d_A) but for a series of ellipses where we keep the H bin fixed at $z = 1.5$ while varying d_A in the redshift range $0.1 < z < 5$. This again intersects the peak FoM when the d_A bin is $z = 0.6$ in the landscape depicted in Figure 5.8. Here we see a distinct narrowing of the parallel lines of the degeneracy direction for d_A from $z = 0.1$ to $z = 0.55$ which impacts favourably on constraining the produced error ellipse. This effect is highlighted in Section 5.3 where the right panel of Figure 5.6 shows the eigenvalue λ_1 for d_A quickly rise to peak in this redshift

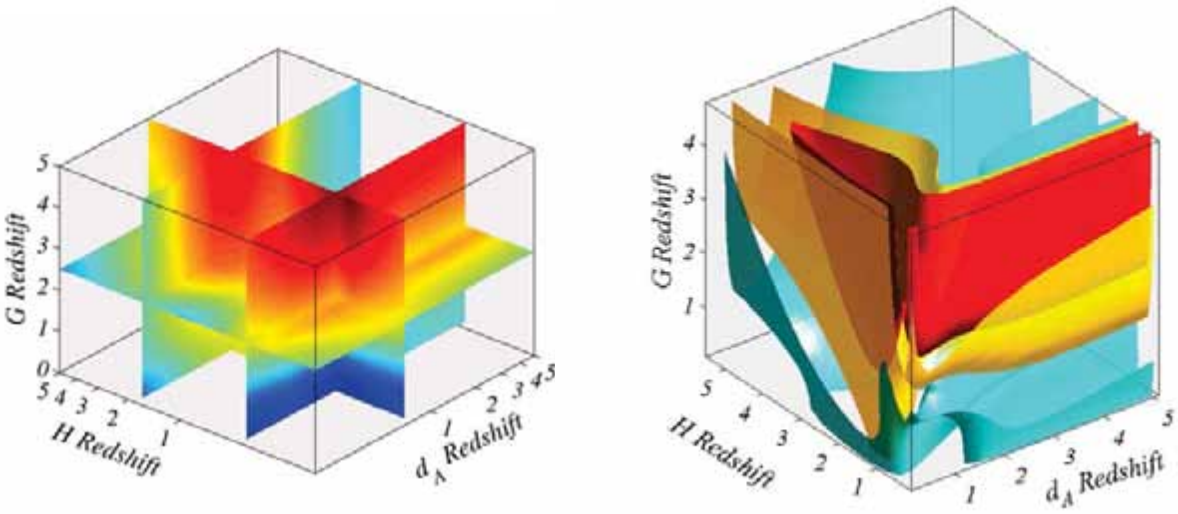


Figure 5.9: **4D Fisher Hypersurface Visualisations** – DETF FoM for the combination of a single measurement each of H, d_A and G (where the growth is normalised to unity at $z = 0$). The redshift of each measurement is allowed to vary, generalising Figure 5.8 by including growth. Slices through this hypersurface, shown in the left panel, at $z_H = 1.67$, $z_{d_A} = 0.67$ and $z_G = 2.50$ show the relationship between the redshifts at which the measurement of the observables are made and the FoM. The colourmap goes from a FoM of 9.5×10^{-4} (dark blue) to 0.14 (dark red). Adding a high-redshift measurement of the growth function tightens the constraints on the dark energy parameters w_0 and w_a, shown by the red ridge of high FoM values. This hypersurface can be illustrated in a complementary way: surfaces of constant FoM are shown in the right panel - ranging from 0.03 (transparent light blue outer surface) to 0.15 (dark red opaque centre surface). Comparing the left and right panels, one notes that the intersection "hotspot" region in the left-hand panel at $(z_H = 0.4, z_{d_A} = 0.67, z_G = 5)$ is contained within in the red iso-surface with the high value of the FoM. This iso-surface extends right down to redshift $z_G \sim 1$, this reiterates the improvement on dark energy constraints when including growth.

range.

5.5 The Effect of Cosmic Curvature On Dark Energy Constraints

The degeneracy between curvature and dark energy has been well studied even for perfect measurements of any single observable such as $H(z)$ [108–112], implying that marginalising over the curvature is important when performing parameter estimation and forecasting constraints on dark energy. The degree to which curvature affects dark energy constraints is shown here as a simple example of Fisher4Cast.

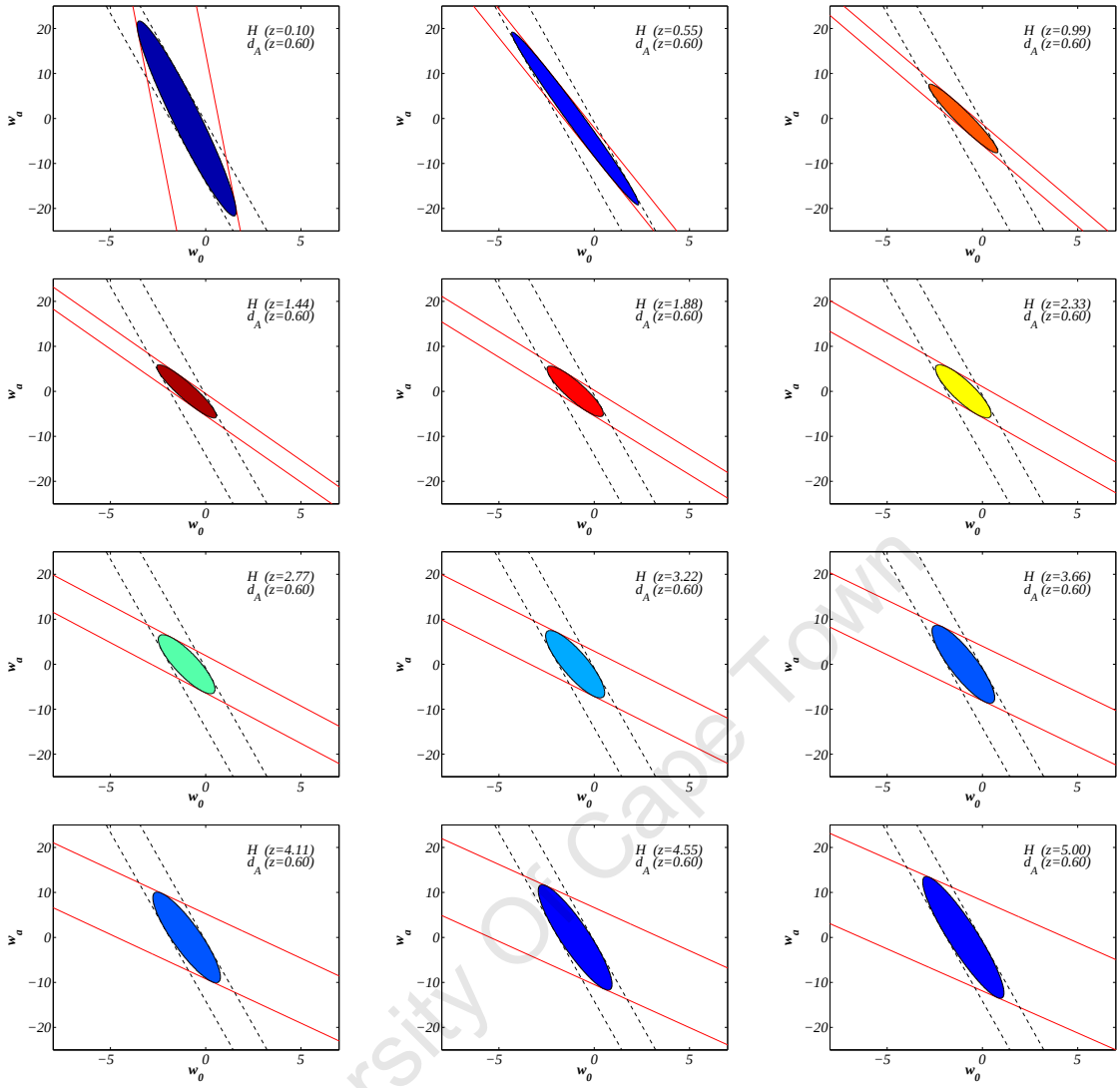


Figure 5.10: **Varying redshift of H** – examples of ellipses corresponding to particular values in the FoM landscape of Figure 5.8, in the particular case where the redshift bins for H are varied while keeping the bin for d_A fixed at $z = 0.6$. The degeneracy direction for d_A (black dashed line) and H (red solid line) are included to help see how their relative orientations contribute to the orientation and size of the combined ellipse. The degeneracy direction of H rotates anti-clockwise with the most rapid rotation experienced initially for the low redshift range of H. After $z = 1$ the rotation of H slows down and no longer is significant. As the H and d_A degeneracy directions become more orthogonal so the resulting constraints improve, yielding a higher FoM. The ellipses are coloured corresponding to their FoM where red indicates the largest (around 0.08), and blue the smallest, FoM.

Figure 5.12 shows Fisher error ellipses for the dark energy parameters w_0, w_a, after marginalising over curvature, as the prior information on curvature is changed from Prior$(\Omega_k) = 10$ (weak) to Prior$(\Omega_k) = 10^6$ (almost perfect) for the observables H, d_A

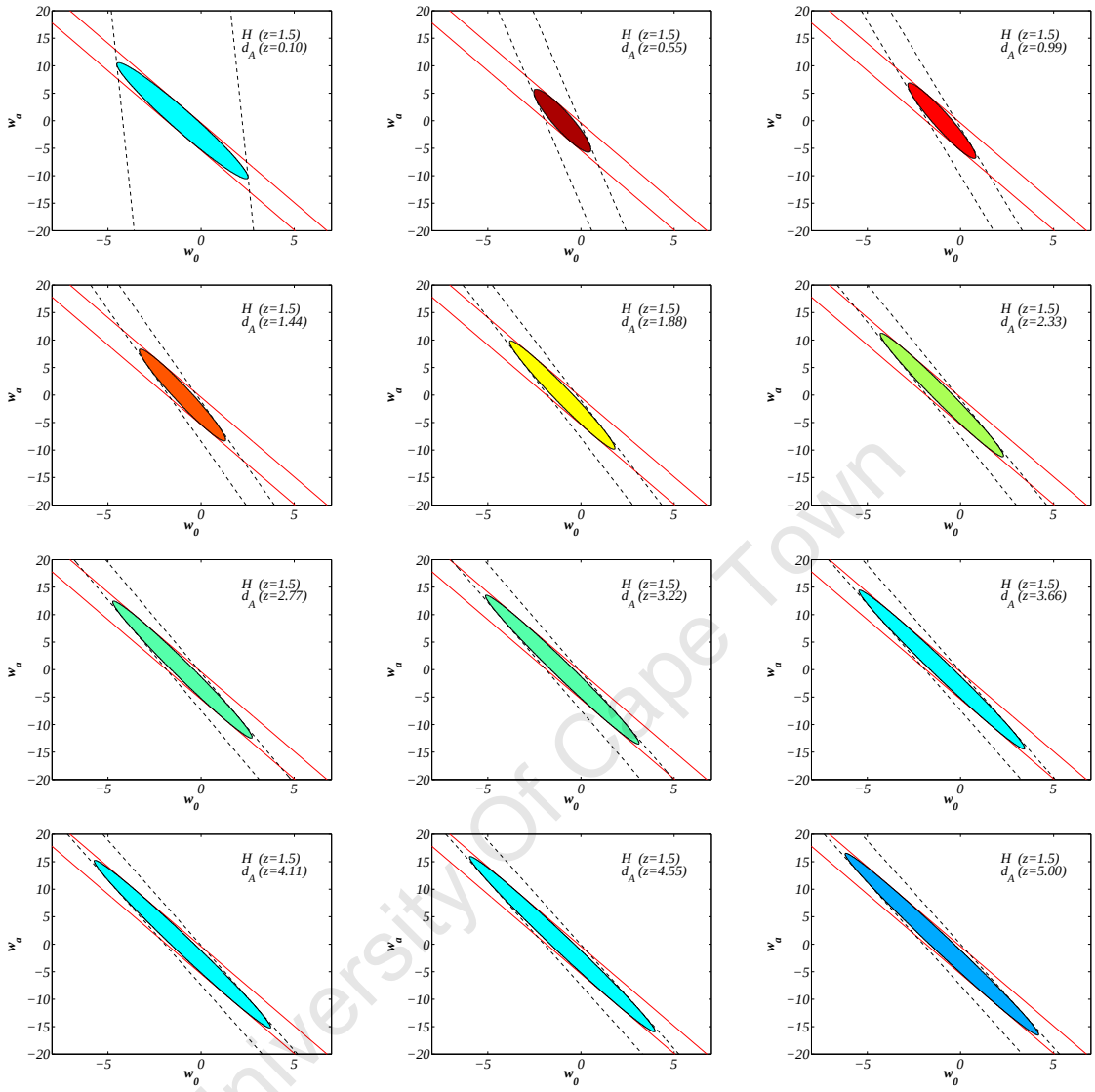


Figure 5.11: **Varying redshift of d_A** – examples of ellipses corresponding to particular values in the FoM landscape of Figure 5.8, in the particular case where the redshift bins for d_A are varied while keeping the bin for H fixed at $z = 1.5$. The degeneracy direction for d_A (black dashed line) and H (red solid line) are included to help see how their relative orientations contribute to the orientation and size of the combined ellipse. The parallel lines of the d_A begin by undergoing a rapid narrowing from $z = 0.1$ to $z = 0.99$. This is followed by a gradual anti-clockwise rotation of the degeneracy direction of d_A which slowly aligns with that of H resulting in progressively weaker constraints, yielding a smaller FoM. The ellipses are coloured corresponding to their FoM where red indicates the largest (around 0.08), and blue the smallest, FoM.

and G considered separately and in combination. While uncertainty in the curvature of the universe (represented by a small prior value on Ω_k) degrades all ellipses, this is much less pronounced when the observables are considered in combination, showing the importance of combining multiple probes of dark energy. For each of

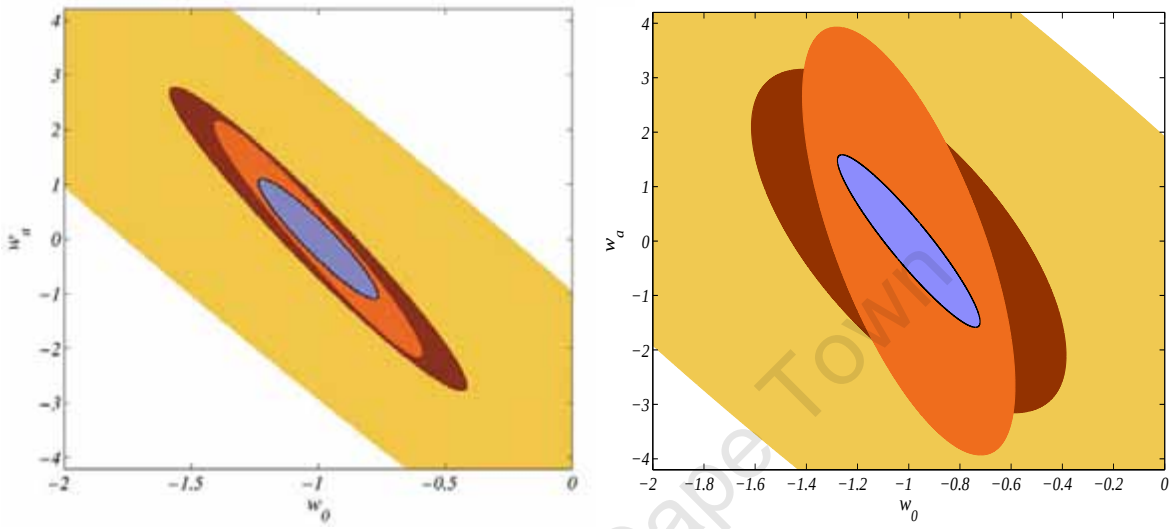


Figure 5.12: **Curvature marginalisation and dark energy constraints [2]** – These two figures show how the weakening of priors on curvature density, from the left panel to the right, result in the DETF FoM being reduced from 3.298 to 1.890 for the inner blue ellipse in the $w_0 - w_a$ plane. The left panel shows the ellipses with a strong Prior($\Omega_k = 10^6$) while the right panel uses a weak Prior($\Omega_k = 10$). Table 5.1 was used for the survey details with an additional 10% measurement on growth at the same redshift as considered for $H(z)$ while the prior on matter density were kept fixed at 100. The fiducial model used was considered flat and the ellipses are given, from largest to smallest, as: $G(z)$ (light brown outer band), $H(z)$ (dark brown filled ellipse), $d_A(z)$ (orange filled ellipse) and the combination of all three (blue filled inner ellipse, outlined in black).

the parameters, Figure 5.13 shows that the constraints are eroded and the ellipse increases in size with the decrease in the prior which expresses our confidence in the flatness of the universe. While the ellipses from the single observables such as $H\&d_A$ show a much greater increase in size with a decrease in the curvature prior, using a combination of parameters is more robust. By combining multiple probes we are able to break the curvature-dark energy degeneracy more effectively. The change in the size of the ellipse directly relates to a change in the Figure of Merit (FoM), or power, of the particular combination of observables. Various FoMs are listed in Sec-

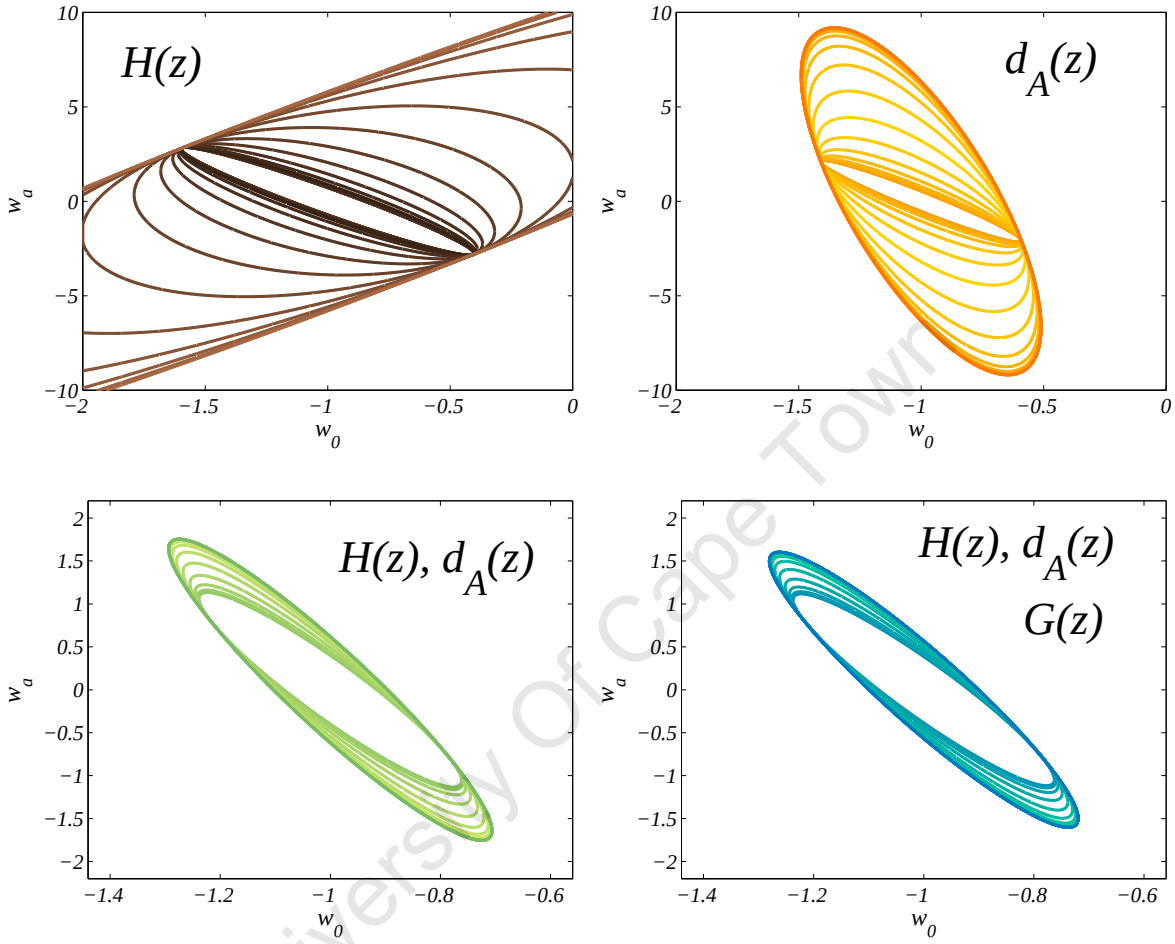


Figure 5.13: **Curvature marginalisation degrades dark energy constraints [2]** – This panel of figures shows how each ellipse produced has progressively less tight constraints on w_0 and w_a as the priors considered for Ω_k become weaker, from 10^6 for the inner ellipse to 10 for the outer ellipse. The survey data used to produce these ellipses is the same as that described in Figure 5.12. The ellipses correspond to measurements used for the Hubble parameter (dark brown curves - top left panel), the angular diameter distance (orange - top right), a combination of the Hubble parameter and the angular diameter distance (green - bottom left) and a combination of all three observables (blue - bottom right).

tion 6.2.3 of the following chapter. In the case of the FoM used by the Dark Energy Task Force (DETF) [66], the area of the ellipse and the DETF FoM are inversely proportional. More information about the DETF FoM and other FoMs available with Fisher4Cast are discussed in Section 6.2.3 in the next chapter.

5.6 Conclusion

Fisher Matrix formalism has a wide scope of application and is the cornerstone of forecasting in modern cosmology being able to produce rapid and insightful results. The need for a software tool to encompass the requirements of students and researchers to have a reliable and modular code that is able to address both the need for simple quick intuitive use and advanced applications led to the development of Fisher4Cast.

This chapter highlights its use starting with a simple cosmological application where the change in orientation and size of the ellipses as a function of redshift was examined. We then studied more novel uses of the Fisher4Cast suite by exploring the landscape of Fisher Matrix cosmology, as illustrated by Figures 5.7-5.9, as well as depicting the effects of curvature on Fisher Matrix forecasts of future cosmological surveys. This forms only a small sample of what can be done with Fisher4Cast and more ideas are addressed in [2].

In the next chapter we turn our attention to the engineering of the code and some of its usefull extensions that have recently been added. These can facilitate more innovative applications and form the building blocks for a more complete and mature research software.

Chapter 6

Fisher4Cast

This chapter is composed of extracts presented in [3] and the Quickstart guide released with the Fisher4Cast code.

Fisher Matrix theory was introduced in Chapter 1 illustrating its value for predicting the propagation of errors for functions of specific parameters. In a cosmological context it is useful for predicting the sensitivity of future surveys and evaluating how errors on measurements (data) map onto the space of errors on inferred cosmological parameters. Examples illustrating the range of applications and versatility of Fisher4Cast was highlighted in Chapter 5.

The aim of releasing the Fisher4Cast suite is to provide the community with a standard, tested tool set for computing Fisher matrix predictions for general observables and survey parameters for use in both research and education. The toolbox design is robust and modular allowing for simple additions and adaptation while keeping the user interface intuitive and easy to navigate. The underlying modular code of Fisher4Cast is completely general and is not specific to cosmology although the default setup for the GUI is intended for cosmology. It provides parameter error forecasts for cosmological surveys given angular diameter distance, Hubble expansion and growth measurements in a general, curved FLRW background. These features were illustrated in the prior chapter with an emphasis on the versatility and applicability of Fisher4Cast.

A common criticism of scientifically released code is that it is poorly documented with little thought given to future extensions, general usability and good coding practices. For this reason we next examine the coding philosophy of Fisher4Cast with a detailed overview of the subfunctions used in the core program. This is highlighted

in the flowchart of the code while the global accessibility of the input and output structures are also documented. This is particularly useful since this forms a basis for future open and modular development to Fisher4Cast.

The code is written in Matlab and can be freely used and modified under the terms of the BSD License[1]. The code is composed of 3000 core lines of code (28 functions) and an additional 800 lines for extensions and extra functionality (extra 25 functions). Both the command line and GUI versions of the code yield plots that are generated directly from the program and can be easily edited in the GUI and saved in a variety of standard formats (eps, jpg, pdf etc...) for inclusion in research publications. There is also a reporting extension which allows for either a text report to be generated or a LaTeX report, which produces the LaTeX syntax for incorporating the input and output in tables, matrices and even a snippet of code for the inclusion of the figure. This allows for rapid and efficient export of the results from Fisher4Cast into a research publication.

The simple start-up procedure and ease of use of the Fisher4Cast suite make it a tool well-suited to both teaching and research purposes. The input to the Fisher4Cast code can easily be changed and adapted, which in turn allows for the generation of varied outputs quickly. This is well-suited to exploring how the specific fiducial models of the parameters, observables and priors all combine to produce error ellipses, and is easily applicable to problems of survey optimisation. A shortened version of the start-up procedure of Fisher4Cast is next outlined to familiarise the reader with its functionality.

6.1 Quickstart Users Overview

This software is written to be run in Matlab (Tested extensively on Linux, Windows and to a lesser extent on Mac OS X). The user needs Matlab installed (Tested on Version 7) to be able to run this code. Free disk space of approximately 5MB and the minimum recommended processor and memory specifications required by the Matlab version being used is suggested. Currently the code is available at [113] or [114].

6.1.1 Getting started

The code can be run from the command line or the Graphical User Interface (GUI). We describe the command line below, and mention how to get the GUI started. For more information on the GUI, please see the manual.

[1]Read http://www.opensource.org/licenses/bsd-license.php

The Graphical User Interface

- Running the GUI

  The GUI can be started from the Matlab editor. The file **FM_GUI.m** must be opened from the directory, and once the file is opened (click on the file icon from within the Command-line interface to open it with an editor) press F5 to run the code. This will open up the GUI screen.

  You can also launch the GUI from the command line by typing:

  ```
  >>FM_GUI
  ```

  This then functions in the same way as using FM run in the command line (as explained in the following subsection).

  For more information on the technicalities of the GUI, see an abridged version of the manual in Appendix C.

- GUI Screenshots

  We include some screenshots of the Graphical User Interface in Figure 6.1-6.3. This shows how different ellipses can be plotted simultaneously along with varying the colour schemes for the GUI and also highlighting how to select alternate FoMs.

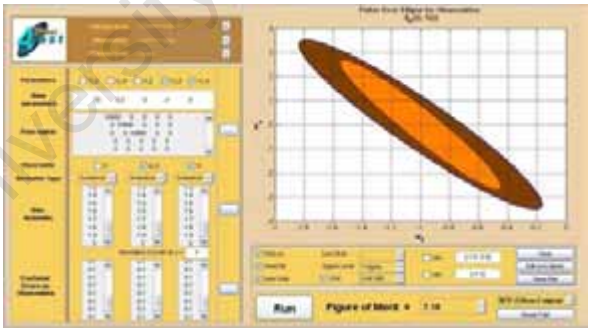


Figure 6.1: **Plotting multiple ellipses on one axis** - using the 'Hold on' multiple error ellipses can be overlaid on one axis. The 'Area Fill' command allows a specific colour to be chosen to fill the error ellipses with. Also shown is the 'Running' window which indicates the code is running to calculate the Fisher ellipses.

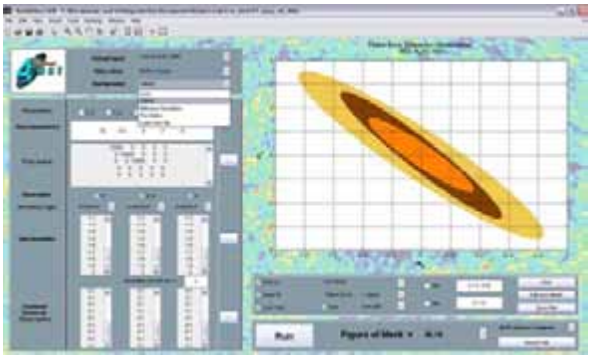


Figure 6.2: **Different background images and colour schemes** - the background images and colour schemes (skins) allow for a fully customisable Graphical User Interface.

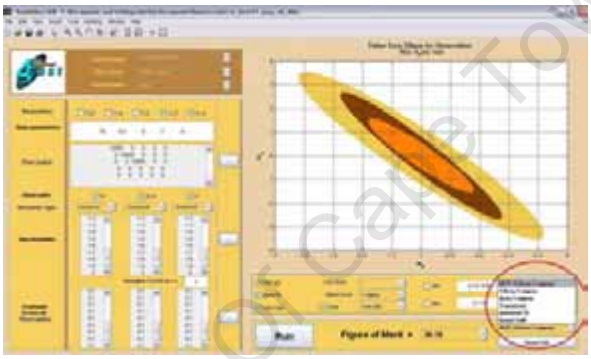


Figure 6.3: **Various Figures of Merit can be plotted** - the drop-down list allows for a choice between various Figure of Merit options.

The Command Line

- Running the code

  Open a current version of Matlab and change the working directory to be the same as the saved directory of Fisher4Cast. To run the code from the command line with one of the standard test input structures supplied, type:

  ```
  >>output = FM_run(Cooray_et_al_2004)
  ```

  This will call the code using the pre-supplied test input data (Cooray_et_al_2004) and then generate an error ellipse plot for the parameters and observables supplied in the chosen input. All the relevant generated output is written to the output structure. You can see the range of outputs to access by typing:

```
>>output
```

and then examine each output individually by specifying it exactly. The details of the output structure are given in Section 6.2.3. For example:

```
>>output.marginalised_matrix
```

will access the marginalised Fisher matrix from the output structure. You can use the supplied input files as a template for generating new input files with user defined parameters and values. All fields shown in the example structures should be included in any user-defined structure.

The code can also be run from the Matlab editor. Once the code is opened (open it from inside the Matlab window), press F5 to run the code. Note that if the code is run from the Editor it will call the default input structure, which is the Cooray_et_al_2004.m file. This is an example file containing input data from the paper by Cooray *et al.* [115] and the output from Fisher4Cast should be an exact match to that of Figure 1 in [115]. Another default input available is Seo_Eisenstein_2003.m [14] which was listed and used in Table 5.1 in the previous chapter.

6.2 Software Engineering and Design

Throughout the code structures are used to allow different sub-parts of the general code access to the data. These structures are defined as global variables, which means that the data contained within them is available to all other functions without being explicitly passed to them. The structures containing information on the input for the code are either defined at the beginning or loaded from a file. All output structures can be saved for later use. The flowchart shown in Figure 6.4 summarises the layout and structure of the code, while the symbols are listed in Figure 6.5. We now discuss the various components of the code in detail. With each subsection we give an extract of the flowchart to show which set of components of the flowchart will next be examined and describe.

6.2.1 Initial Processing

FM_run

FM_run.m is the general wrapper of the code and the descriptions listed below document the components and outline of this function. In order to make the code

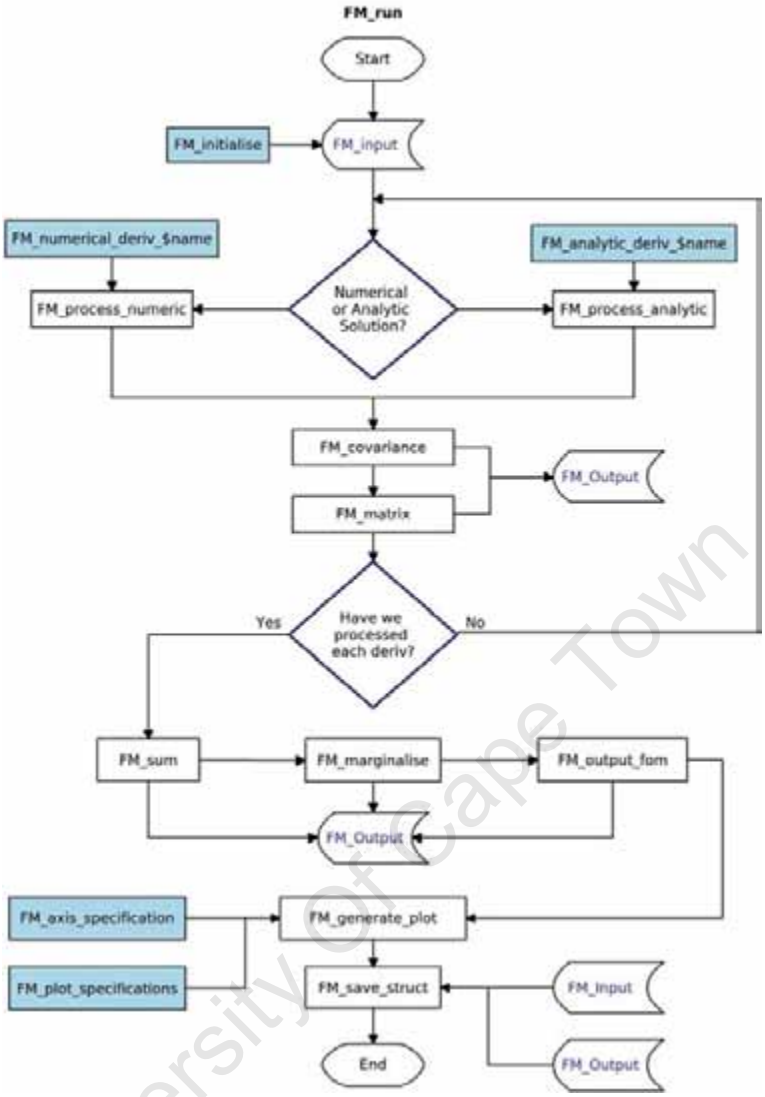


Figure 6.4: **Fisher4Cast Flowchart FM_run.m** – Here the outline of the core code of Fisher4Cast is shown listing the processes, decisions and storage of data in order of execution. For a key to the symbols see Figure 6.5.

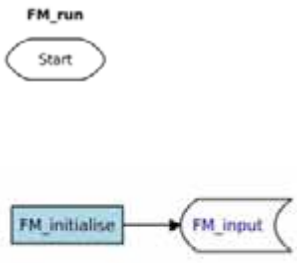
clear and easy to edit, all the main processes are called from this core function. Links to separate functions called from **FM_run.m** are documented in the comments and preamble of the code. As explained in Section 6.1.1 only a single function is needed from the command line to run Fisher4Cast. A specific input can be passed with **FM_run.m** but if no input is supplied then the **Cooray_et_al.m** structure is used by default.

FM_initialise

This function initialises the input used throughout the processing of the Fisher4Cast code. The values, names and areas of interest are specified here. It is called by **FM_run.m** to set the initial values for the **input** structure, which is then globally accessible to all other parts of the code. Examples of the default initialising functions provided are **Cooray_et_al_2004.m, Seo_Eisenstein_2003.m** [14, 115]. The default input structures should be used as templates for future custom input structures since the variables listed in the input files are expected by the code. The entries are as follows:

- **input.function_names** - A cell of strings containing the specific filenames of the *analytical derivatives*. Note in the coded cosmological example that no analytical derivative function is specified for the growth function, derivatives are only taken numerically.
- **input.observable_names** - A cell of strings specifying the names of the observables.
- **input.observable_index** - A vector of the indices corresponding to the observable_names of interest, e.g. [2 3] would imply that we are considering the second and third observables listed in observable_names.

Figure 6.5: **Component Key for Fisher Flowchart** – This key shows the symbols contained in the flowchart, Figure 6.4. They are from, left to right, the begin and terminate indicator; a simple processing function which would generally return an output; an if statement, for or while loop; an input process function designed to be edited and changed as per the user specifications and lastly a stored structure for either input or output and passed globally for use throughout the code.



- **input.data{i}** - These are the row vectors of the data for each of the respective observables (indexed again from beginning to end by i). In the cosmological example **input.data{1}** would be the redshifts of the Hubble parameter measurements, for example.

- **input.parameter_names** - A cell of strings containing the corresponding names of the parameters which can be included for consideration to generate Fisher ellipses.

- **input.base_parameters** - A row vector of the parameter values (they must be specified with the same order as the parameter_names vector) for the assumed base input model. The Fisher Matrix is taken around this fiducial model.

- **input.prior_matrix** - The prior matrix for the parameters taken from previous surveys etc. where large values correspond to good prior information to smaller values which indicates less prior knowledge. The order of the matrix columns and rows correspond to the respective parameters listed in parameter_names. If no prior knowledge is available then the values can either be set to zero or a checkbox on the GUI is available to disregard the priors.

- **input.parameters_to_plot** - A row vector of the indices of the specific parameters that are to be plotted. If one index is checked then a likelihood function for that parameter is plotted while if two are specified then an error ellipses is plotted. Selecting more than two parameters will produce an error message, as Fisher4Cast is only coded for up to 2-dimensional error contours.

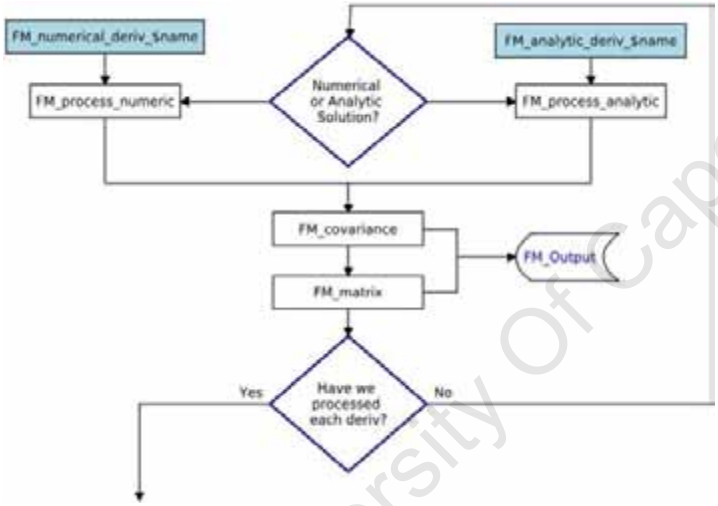
- **input.num_parameters** - This is a value that doesn't doesn't need to be specified and is derived from the number of parameters which are considered in **parameters_to_plot**.

- **input.num_observables** - This is a value that doesn't need to be given and is derived from the number of observables under consideration in observable_index.

- **input.error{i}** - The fractional error on the data from the observables ($\sigma_{\mathbf{X}^\alpha}/\mathbf{X}^\alpha$). It is key that there are as many error entries as there are observables being considered (i.e. **input.error{1}** gives the error on the measurements of the Hubble parameter, measured at **input.data{1}**). In the case of uncorrelated observables only a row vector is required and this will be converted to a diagonal covariance matrix. If the observables are correlated then the full covariance matrix is to be specified.

- **input.numderiv.flag** - A logical entry is expected (1-True ; 0-False) to specify if a numerical derivative is required for each appropriate observable. This corresponds to the order of observables as first listed in **input.observable_names**.
- **input.numderiv.f** - This is a vector of single string entries which lists the function names of the numerical derivatives that are to called for each respective observable e.g. A numerical derivative for the observable of growth of structure is required and is specified in **input.numderiv.flag**, the function to be called to preform the numerical derivative for growth is given in **input.numderiv.f** as **g.m**.

6.2.2 The Derivative Loop



The code now runs various operations in a loop over the observables. The **input.numderiv.flag** for each observable is checked to determine whether numerical or analytical derivatives are to be used. Both the analytical and numerical derivatives return a matrix of derivatives for all the parameters and observables as well as a vector of the function evaluated at the data points specified. The specific details of the numerical and analytical derivative codes are next dealt with. Once the selected derivative process is completed, the relevant output is stored in the **output** structure which is globally available to any other function.

Numerical Derivatives

The numerical derivative code will calculate the numerical derivatives of any function (say **g.m**) provided that the function is specified as a function of the input parameters

(i.e. $g = g(\mathbf{d}, \theta_{\rm A}, \theta_{\rm B}, ...)$), by calling on **FM_process_numeric.m** which in turn calls **FM_num_deriv.m** and passes it the name of the function which have been selected to take derivatives of.

The standard numerical derivative algorithm used is known as the complex-step method [116]:

$$\frac{\partial g}{\partial \theta_{\rm A}} = {\rm Im}\left[\frac{g(\mathbf{d}, \theta_{\rm A} + ih, \theta_{\rm B}, ...)}{h}\right],$$

where Im represents the imaginary part of the argument, and $i^2 = -1$ as usual. This method is a second order accurate formula and is not subject to subtractive cancellation. Unlike the finite-difference method an arbitrarily small step-size can be chosen and therefore the complex-step method can achieve near analytical accuracy.

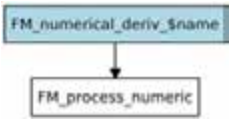
In addition, the simple double-sided central derivative is coded in the **FM_num_deriv.m** function. In order to use this algorithm the user must change the method field inside the derivative function from 'complex' to 'central'. In this case the gradient is then calculated as

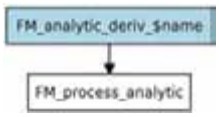
$$\frac{\partial g}{\partial \theta_{\rm A}} = \frac{g(\mathbf{d}, \theta_{\rm A} + h, \theta_{\rm B}, ...) - g(\mathbf{d}, \theta_{\rm A} - h, \theta_{\rm B}, ...)}{2h}.$$

This is then iterated until the gradient converges for the parameter. Note that the convergence criterion is quite stringent and an error message will result if there are possible convergence issues. However this criterion can be relaxed by changing the settings in the **FM_num_deriv.m** code.

Once the derivatives are saved the Fisher Matrix must be calculated for this observable. This is accomplished by first calculating the data covariance matrix for the observable. This is accomplished with **FM_covariance.m** which is passed the function value and the index specifying the observables under consideration. The code checks if the error entry is a covariance matrix (in the case of correlated observables) or a vector in the uncorrelated case. The covariance matrix is then calculated by multiplying the variance with the function value at the data points considered.

FM_matrix.m subsequently produces a Fisher Matrix (F) from the covariance matrix (C) and the derivative matrix (V) using matrix multiplication as $F = V^{\rm T} C^{-1} V$.





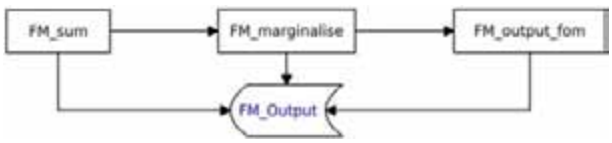
Analytical Derivatives

The analytical derivatives must be specified for each particular application. If one knows the analytical form of both the function and of the Fisher derivatives, one can include these functions explicitly. The only conditions on these functions are that they must be of the form $g = d(\mathbf{d}, \boldsymbol{\theta})$ and must return as output a matrix of Fisher derivatives $\partial g/\partial \boldsymbol{\theta}$ and a vector of the function itself evaluated at the data points $\mathbf{d}$ given in the **input.data{i}**. These derivative functions are supplied for the Hubble parameter and angular diameter distance as **FM_analytic_deriv_1.m** and **FM_analytic_deriv_2.m** respectively. The Fisher derivatives of the angular diameter distance with respect to the cosmological parameter Ω_k must be taken as Taylor series expansion when $\Omega_k \to 0$ (see Appendix B.1 for the full set of derivatives in Fisher4Cast).

As in the numerical derivative case, once the derivatives are saved the Fisher Matrix must be calculated for this observable. This is done by calculating the data covariance matrix for the observable in **FM_covariance.m** which is passed the function value and the index α of the observable. The code checks if the errors are given as a covariance matrix (in the case of correlated observables) or a vector in the uncorrelated case. It then calculates the covariance matrix by multiplying the variance with the function value at the data points considered.

FM_matrix.m then produces a Fisher Matrix (F) from the covariance matrix (C) and the derivative matrix (V) using matrix multiplication as $F = V^{\mathrm{T}} C^{-1} V$.

6.2.3 Final processing



FM_sum.m collates all the derivative matrices from the previous steps and sums them together to form a full Fisher Matrix. The individual Fisher matrices for each observable are added to the prior matrix (as specified in the **input** structure). This

complete Fisher Matrix is general and is assigned to the **output** structure for future reference.

FM_marginalise.m produces a marginalised Fisher Matrix (say $\widetilde{F}$). It takes the parameters of interest (specified as **parameters_to_plot** in the **input** structure) and shuffles the Fisher Matrix into a block form. It then performs matrix multiplication on the blocks to produced the marginalised Fisher Matrix, which is also assigned to the output structure.

FM_output_fom then produces the appropriate error for the likelihood case and a range of Figure of Merits (FoMs), listed below, for the case of an ellipse (e.g. if the length of **parameters_to_plot** in the **input** structure is two then an ellipse will be plotted). These FoM are then stored in the **output** structure. Fisher4Cast includes the standard FoMs as well as some new ones available through the GUI and command line. The FoM are only calculated for the parameters chosen in **input.paramater$_t$o$_p$lot** and not the full 5-D matrix. It should be noted that some of the FoMs are only defined for the error ellipse in the $w_0 - w_a$ plane, where w_0, w_a are the coefficients in the Chevalier-Polarski-Linder (CPL) [8, 9] parameterisation of the equation of state for dark energy (see for e.g. [66]). We briefly outline the FoMs used in Fisher4Cast:

- DETF
  The FoM in the Report of the Dark Energy Task Force [66] is defined to be the reciprocal of the area of the 2σ error ellipse in the $w_0 - w_a$ plane of the CPL dark energy parameterisation [8,9]. This is, equal to $\det(F^{1/2})/(\pi\sqrt{6.17})$. Unfortunately the DETF report does not appear to use this definition, and instead quotes $\det(F^{1/2})$, which is the inverse of the 1σ ellipse in units of the area of the unit circle. Because of the benefits of the geometric interpretation Fisher4Cast returns the true inverse area of the 2σ ellipse. To convert from one DETF FoM to the other, one should multiply the Fisher4Cast DETF output by $\pi\sqrt{6.17} \simeq 7.8$

- $\text{Area}_{1\sigma}^{-1}$
  This Figure of Merit is the reciprocal of the 1σ error ellipse area in the parameter plane currently plotted, i.e. $\det(F^{1/2})/(\pi\sqrt{2.31})$

- $\text{Area}_{1\sigma}$
  Simply the inverse of the previous FoM.

- Tr**C**

This FoM is defined as the trace of the covariance matrix of the data, $\mathbf{C} = \mathbf{F^{-1}}$, estimated as the inverse of the marginalised Fisher Matrix. This FoM is simply the sum of the squares of the marginalised errors on each parameter.

- $\sum_{\mathrm{AB}} C_{\mathrm{AB}}^2$
  This FoM is defined as the sum of the squares of the entries of the whole covariance matrix, $\mathbf{C} = \mathbf{F^{-1}}$. Unlike the previous definition this FoM is sensitive to the off-diagonal components of the covariance matrix as well as the diagonal components.

FM_output

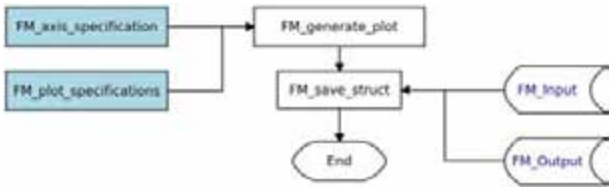
The data from all parts of the code are saved in the **output** structure. The structure formalism in Matlab means that each '.' indicates a further sub-level in the structure. Entries in the structure are of mixed type (i.e. **output.function_value** is a cell of vectors, one for each observable, while **output.function_derivative** is a cell of matrices of derivatives, again with one matrix of derivatives for each observable). By the end of the execution of **FM_run.m** the **output** structure should have the following entries:

- **output.function_value** - This is a cell which contains a set of vectors for each of the observables considered. So if only one observable was selected in the input then only the corresponding output.function_value will contain data while the other vectors will remain empty.
- **output.function_derivative** - This cell now contains matrices of the Fisher derivatives for the observable. Again, the entries of the observables not selected will result in empty matrices.
- **output.data_covariance** - This cell contains the calculated data covariance matrix corresponding to each of the observables considered.
- **output.matrix** - This cell contains a separate Fisher Matrix for each of the observables considered.
- **output.summed_matrix** - This cell contains the sum of the Fisher matrices for each observable, and the prior information matrix, if included.

- **output.marginalised_matrix** - The marginalised Fisher Matrix given here depends on which parameters are of interest in each run of the code. The marginalisation via matrix multiplication is outlined in Section 6.2.3.

- **output.fom** - This vector contains either a single entry (1σ error), in the case where a one-dimensional likelihood function a parameter $\theta_{\rm A}$ (for example) is being considered or an array of different FoMs when an ellipse of two parameters is being plotted. These are each explained in the previous section above.

6.2.4 Generating plots



FM_generate_plot calls either **FM_plot_ellipse.m** or **FM_plot_likelihood.m** depending on whether a $1-$D likelihood or an ellipse is required (whether one or two parameters are specified in the **parameters_of_interest** field in the **input** structure). The style of the plot is controlled by the **FM_plot_specifications.m** file, which controls variables such as the line style, the colour of the lines, the resolution of the grid and the contour level (for example $1\sigma, 2\sigma$). Similarly the file **FM_axis_specifications.m** controls the x and y labels and the range of the plot that will be generated.

Lastly **FM_save_struct** is called to save the **input** and **output** structures with a user specified filename. One could invoke this function from the command line:

```
>>FM_save_struct('saved_filename',input,output)
```

where **input** and **output** correspond to the structures that are being saved as **saved_filename-01-Nov-2008.mat**. The date on which the structure is saved is appended to the end of the filename. If a filename is not specified then the default name of **FM_saved_data** is used. It is important to note that the function overwrites existing files with the same name and date and no warning is given. Care should thus be taken to ensure different names are specified when saving important data on the same day. The structures can be loaded once again by issuing the following command:

```
>>load('saved_filename-01-Nov-2008')
```

this will make the previously used **input** and **output** structures available in the current session. Finally we provide a global view of the structure of the code in Figure 6.6.

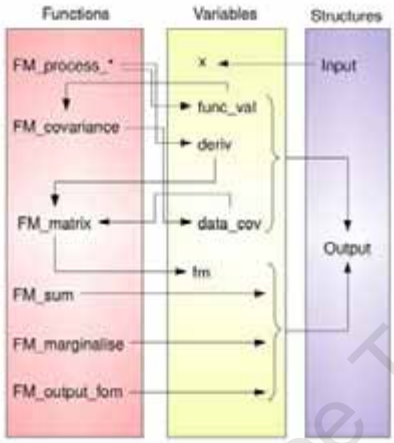


Figure 6.6: **Linking of functions for FM_run.m** – The relationships between the various functions, variables and structures of the code are illustrated here. The vertical bars indicate the functions, variables and structures respectively while the arrows illustrate how a function may produce a variable and how that variable is in turn stored in a structure (left to right). Alternately the arrows can also show how a variable is retrieved from a structure and (possibly) used in a function. The order from top to bottom shows the chronological order in which the functions, variables and structures are called **FM_run.m**.

6.3 Extensions

The general philosophy of Fisher4Cast was to make it as easy as possible to mould and extend to the needs of a general user. In line with this philosophy we have introduced extensions as a means to add functionality and customisation to the existing Fisher4Cast suite. As a design philosophy for future extensions we envisage that extensions do not alter the core functions of the code but rather access the input, output or other modular core-functions. This will enable a large community of con-

tributors to add and make available their own specific extensions while ensuring that the robust design features of the core code remain intact.

An important element in ensuring the success of shared extensions is that all contributors have a good appreciation of the structures used in Fisher4Cast while also documenting and commenting their code thoroughly, including the purpose of the extension, the required input and output produced, which files or structures the extension interacts with from Fisher4Cast and whether it is run from the command line or GUI. The latest extensions included in this release are listed below.

6.3.1 Reporting Features for the Fisher4Cast Suite

Two extension modules have been included to provide reports of the **input** and **output** structures during a run of Fisher4Cast. These reports can either generate an ASCII text file (.txt) or a LaTeX file (.tex) which detail all the input and output produced by Fisher4Cast.

In the case of the LaTeX reporting function the resulting .tex file can be compiled using LaTeX to produce a Postscript file (.ps) or Portable Document Format file (.pdf). This allows for a more polished presentation of the results generated from Fisher4Cast. It also includes a figure of the ellipse or likelihood plot which is embedded in the document. The additional benefit to generating a document in .tex format is that one can cut-and-paste the LaTeX formatted syntax of the figure or any of the tabulated data for easy inclusion in an article or document containing the results from a run of Fisher4Cast. These reporting features are accessible through the Graphical User Interface, by clicking the scroll-down menu bar labelled 'Saving Features'. This opens a dialog box which in the case of the text report prompts the user for the .txt filename that it should be saved as. Upon choosing a LaTeX report, two dialogue boxes are opened and the user is prompted for the names of both the .tex file and the .eps file for the figure.

These two extensions can just as easily be called from the command line. To generate a text report one uses the functions **FM_report_text.m**. The user is required to supply at least an input structure to generate a report. This input structure can either be a default input structures, e.g. **Cooray_et_al_2004.m**, or a user customised input. The function **FM_report_text.m** then calls **FM_run(input)** with the same supplied input which then produces the relevant output structure. Both the input and output used and generated from Fisher4Cast are then recorded in the report. The user can also specify a filename to save the report as (if no .txt extension is supplied one will be added automatically). A default name of 'Fisher4Cast_Report-

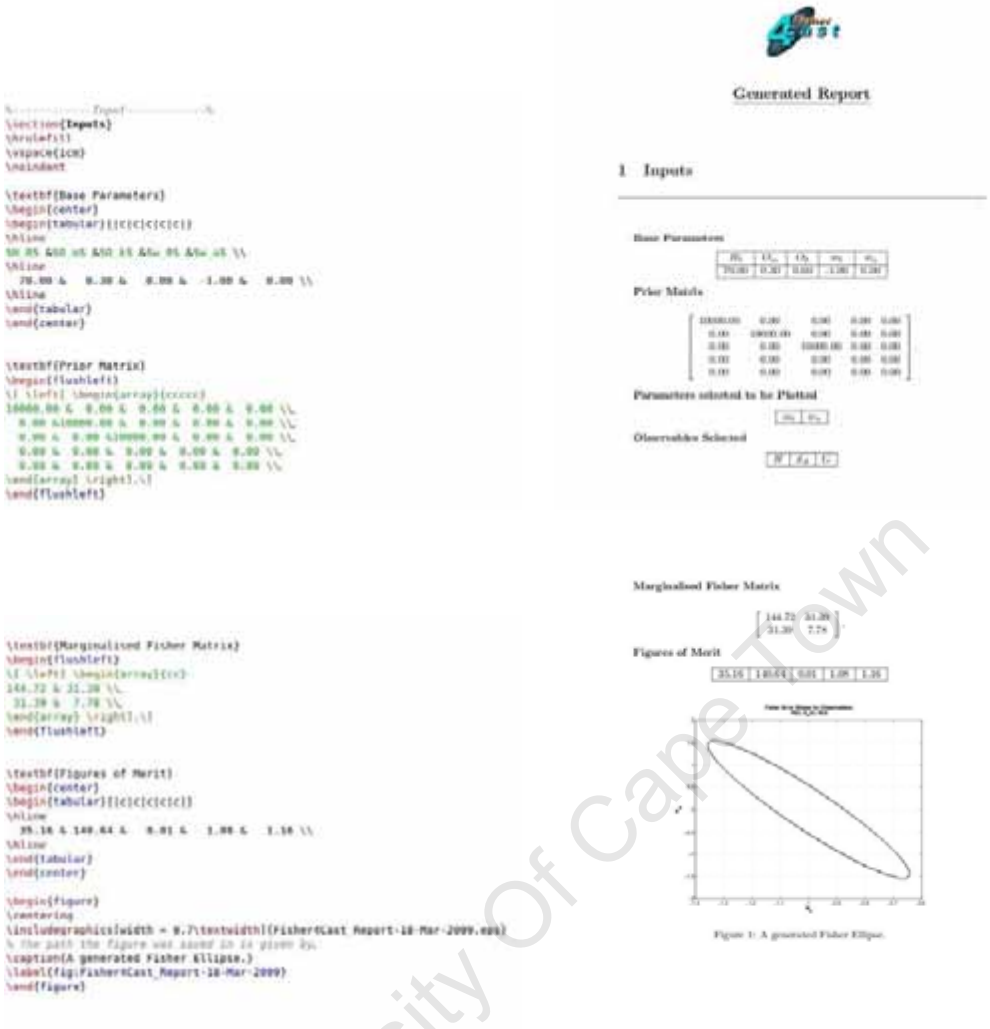


Figure 6.7: **LaTeX Report** – The left panel shows excerpts from the .tex file that was generated from the LaTeX report while the panel on the right shows extracts from the corresponding .pdf file that was generated from the .tex file.

Day-Month-Year.txt' will be used, should no name for the report be specified, where the Day-Month-Year are the date on which the report was generated. For example the command:

```
>>FM_report_text(input,'report_name')
```

will generate a report with the name, 'report_name.txt', as described above. If the same report_name is used, the previous report will be overwritten without warning. Please specify a unique report_name to ensure the report is correctly saved.

Finally there is an option of including a specific output structure in the report function. This is useful when generating the report from the GUI, but care should be

taken when using this option in the command line, as one runs the risk of generating a report where the input and output are not appropriately related. In other words:

```
>>FM_report_text(input,'report_name',output)
```

generates a report as before with the name, 'report_name.txt', using the input supplied and *assuming* that the given output is associated with the respective input.

Much the same as the text report, the LaTeX report is called using **FM_report_latex.m** and requires at least an input structure. The filename the report is to be saved as can also be specified (either with or without the .tex extension). The LaTeX report includes the EPS figure generated from Fisher4Cast. As a default the figure will be saved with the same name as the .tex, except an .eps extension. The default name of 'Fisher4Cast_Report-Day-Month-Year.tex' is used, should no report name be specified. The commands:

```
>>FM_report_latex(input,'report_name')
```

generate a report with the name, 'report_name.tex', and a figure with the name 'report_name.eps', where the names overwrite any existing files of the same name. Additionally, one can use a specific figure in the report with the command:

```
>>FM_report_latex(input,'report_name','use_fig')
```

In this case there is of course no guarantee that the figure and the output from Fisher4Cast agree.

As in the case of the .txt report, one can specify the output structure directly with:

```
>>FM_report_latex(input,'report_name','use_figure',output)
```

which generates a report as before with the name, 'report_name.tex', using a figure called 'use_figure.eps' where the output and figure are assumed to be associated with the respective input supplied.

6.3.2 Obtaining Baryon Acoustic Oscillation Errors from Survey Parameters

Two modules are included that calculate errors on the Hubble parameter and angular diameter distance in BAO surveys. The input for these modules can be supplied in an input structure detailing the survey parameters. The provided codes are extensions to Fisher4Cast, and should be placed in the same folder as the main code suite so that they can access the required elements of the Fisher4Cast suite.

The first of these extensions, **EXT_FF_Blake_etal2005** uses the fitting formulae of Blake *et al.* [13] to calculate the errors on the Hubble parameter and angular diameter distance given certain survey specifications, such as the survey area and the redshifts used for the measurements of H and d_A. These are either given as central redshift bins or as the edges of redshift bins along with the galaxy number density.

The module **EXT_FF_SeoEisenstein2007** also computes the errors on the Hubble parameter and angular diameter distance using the prescription set out in [117] and the sound horizon scale as given in [118]. This module contains a wrapper to call the Matlab version of the **C** code of Seo & Eisenstein [119]. In this module the code does not need to be in the same directory as the Fisher4Cast suite, and runs completely independently of Fisher4Cast.

In both of the above modules default input surveys are supplied. More specific details of the inputs, outputs and code structure can be reviewed in the Appendix C.4. Given the growing importance of the BAO surveys this extension will prove very usefull in both quickly generating results from planned survey specification and in addition be able to provide insight into future survey optimization and design.

6.4 Conclusion

There exists a need for a standardised software package that is specifically designed for Fisher matrix applications. In this chapter we illustrate how Fisher4Cast meets this niche by highlighting the design philosophy. Central to the software engineering principle is that the code is opensource, general, modular and employs professional programming standards. An overview of how these goals where achieved are explained.

Its modularity and object oriented style design is illustrated through the use of flowcharts and an in depth description of each subfunction used in the **FM_run.m** code which is at the heart of Fisher4Cast. There is extensive documentation in the form of the User Manual [3], Quickstart Guide and Readme.txt file (packaged with the download file for Fisher4Cast). The benefit of a reference of this nature is two fold, it enables end users to quickly learn and become familiar with the functionality of the code. Secondly this allows developers to have an overview of the design philosophy along with details of the function and how they can be called and integrated into new code.

This lays a foundation for extensions and further development in the future. Two newly developed extensions are described showing the ease with which it is possible

to build on the current core code of Fisher4Cast. The two extensions lend additional functionality to the cosmological applications. The first being a reporting extension which allows for either a text or a LaTeX report to be generated, which includes the LaTeX syntax for direct incorporation into research publication (including figures). A second extension calculates the errors for the Hubble, H, and angular diameter distance, d_A, from given specifications for a BAO survey. One of two methods can be selected for this calculation, either using the Blake *et al.* [13] module or Seo & Eisenstein [119] module. This extension can be used to both examine the effectiveness of constraining parameters as well as a means to optimize the design of any future planned BAO surveys.

Appendix A

Optimization - Additional Plots

This appendix contains figures that supplement the discussion in Section 3.2.3 and 3.3.1. Figures A.1 and A.2 show the results of varying the proposed step size, σ, for the MCMC algorithm applied to the Rosenbrock and Griewangk test functions. This does not result in significant enough gain, in any of the cases, to outperform Hybrid and is thus not a viable alternative strategy. Tuning σ also proves to be a very time consuming practice, in particular as higher dimensional problems are considered.

Figures A.3 and A.4 show the results of the CPU running time in seconds for MCMC, SA, Hybrid, MCMC-MTM and Hybrid-MTM applied to the Rosenbrock and Griewangk test functions. In all cases Hybrid-MTM converges in the shortest time to the lowest FoM. The Hybrid-MTM results for the 20-50D Griewangk test problem are particularly good showing a clearly superior rate of convergence.

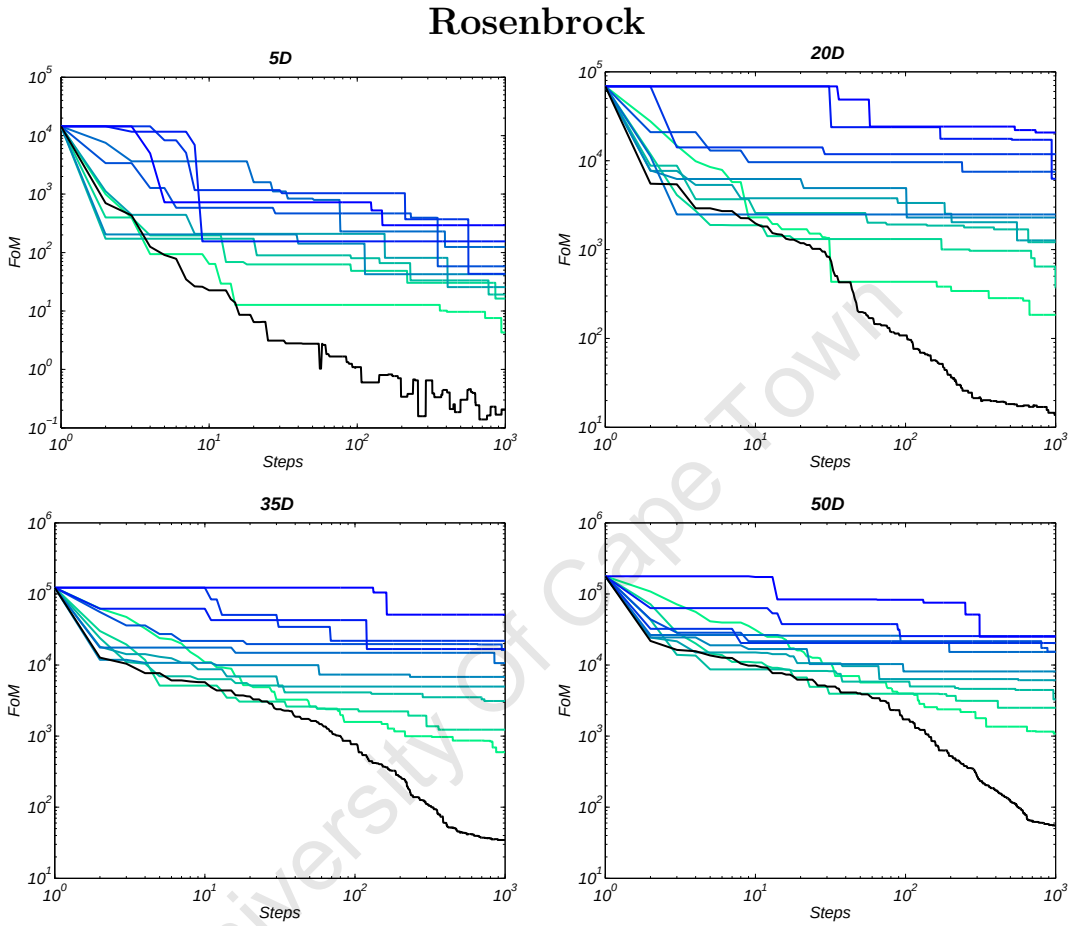


Figure A.1: **Varying MCMC $\boldsymbol{\sigma}$** – The σ is the proposed step size for the MCMC algorithm and is randomly selected from the proposal distribution which is assumed to be a multivariate normal distribution. Illustrated above the σ is varied by a factor from 1 (lightest blue) to 10 (darkest blue) for the MCMC algorithm. In all cases an increase in σ does not help the rate of convergence and only results in an earlier saturation at a higher FoM. Hybrid outperforms all variations of MCMC across the dimensions considered.

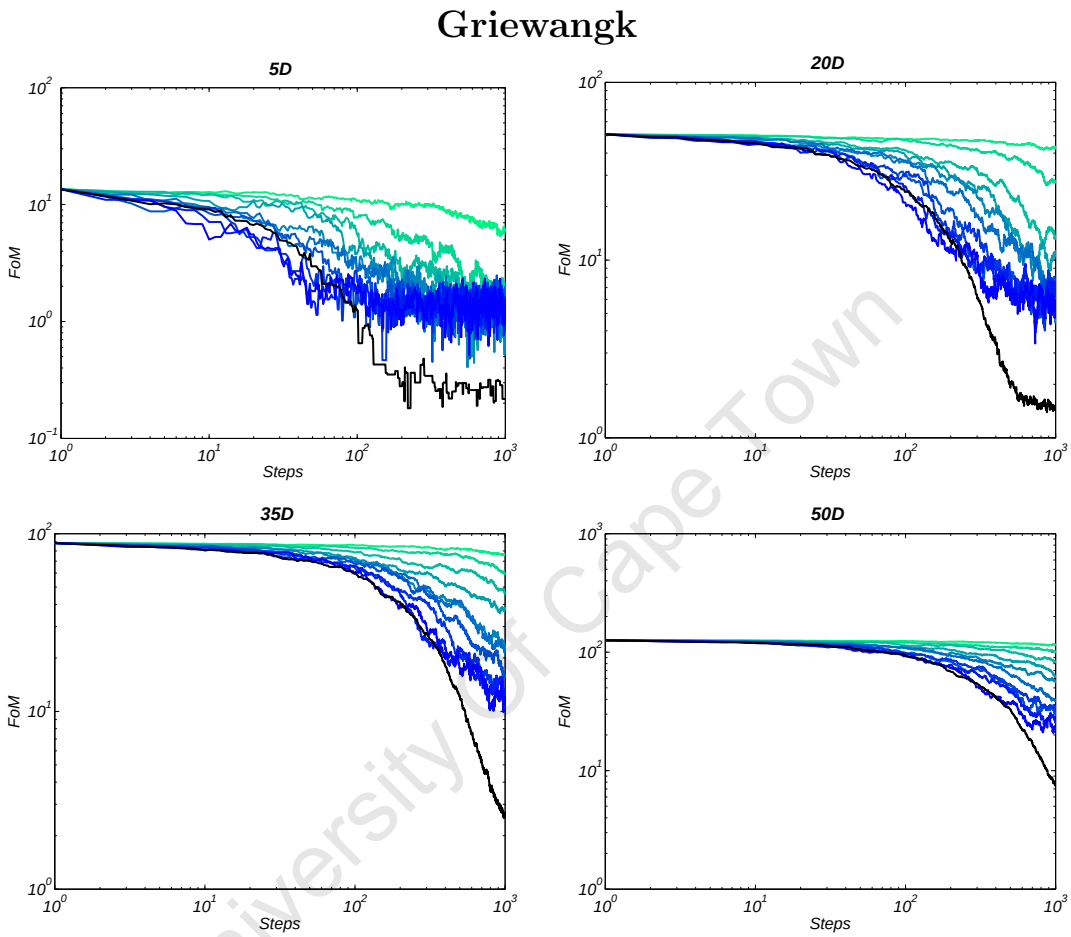


Figure A.2: **Varying MCMC $\boldsymbol{\sigma}$** – The σ is the proposed step size for the MCMC algorithm and is randomly selected from the proposal distribution which is assumed to be a multivariate normal distribution. Illustrated above the σ is varied by a factor from 1 (lightest blue) to 10 (darkest blue) for the MCMC algorithm. In all cases an increase in σ shows a faster initial rate of convergence but does not manage to outperform Hybrid which consistently produces a lower FoM across all the dimensions considered.

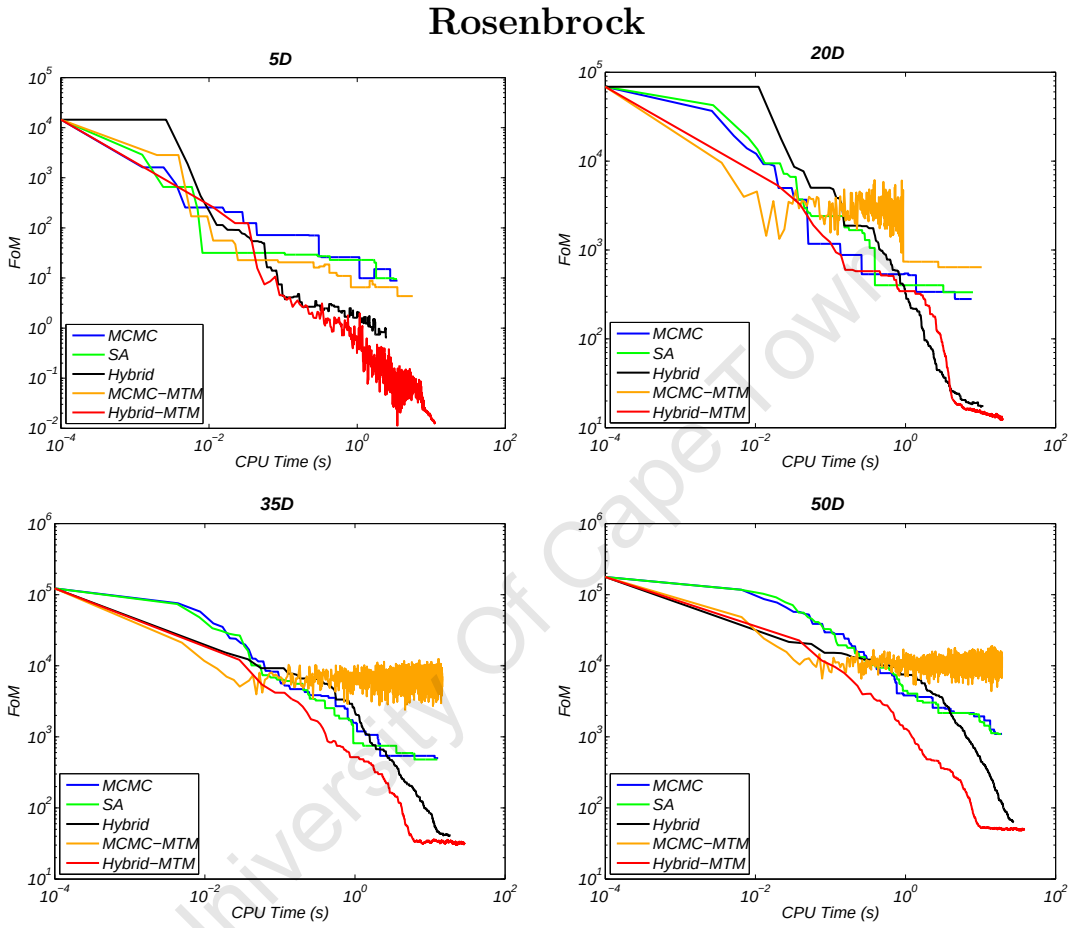


Figure A.3: **CPU Time** – Here a comparison between MCMC, SA, Hybrid, MCMC-MTM and Hybrid-MTM is done on the Hyperboloid test problem to record how the FoM changes as a function of CPU time recorded in seconds. Both Hybrid and Hybrid-MTM have very comparable performance but in all cases Hybrid-MTM produces the lowest FoM in the shortest time.

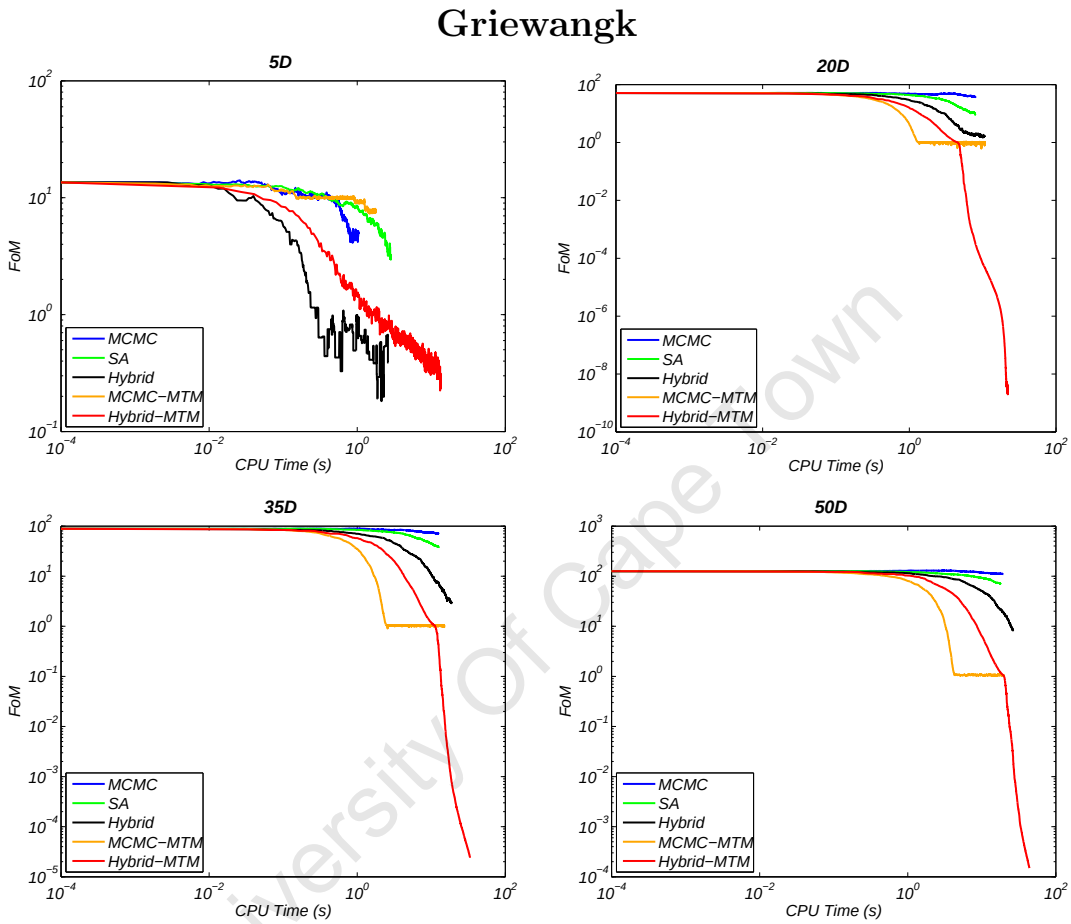


Figure A.4: **CPU Time** – Here a comparison between MCMC, SA, Hybrid, MCMC-MTM and Hybrid-MTM is done on the Hyperboloid test problem to record how the FoM changes as a function of CPU time recorded in seconds. As higher dimesions are considered so Hybrid-MTM's performance improves compared to the other algorithms. Hybrid-MTM impressively outpreforms the other algorithms, producing an FoM 10^4 times smaller than it nearest rival in the same time for the case of the 50D test problem.

Appendix B

Fisher4Cast Applications - Analytic Derivatives

B.1 Fisher Derivatives for $H(z)$ & $d_A(z)$

We present the analytical Fisher derivatives of the Hubble parameter, $H(z)$, given as Eq. (5.2) and the angular diameter distance, $d_A(z)$, given as Eq (5.5) with respect to the cosmological parameters $(H_0, \Omega_m, \Omega_k, w_0, w_a)$, where w_0, w_a are the CPL dark energy parameters, and assume the forms $f(z)$ and $E(z)$ as given in Eqs. (5.4) and (5.2) respectively. In all cases the derivatives are taken in a general Friedmann-Lemaître-Robertson-Walker background without assuming flatness.

B.1.1 The Hubble parameter

As the Hubble constant, H_0 only appears as a multiplicative term in $H(z)$, the Fisher derivative of the Hubble parameter with respect to the Hubble constant H_0 is simply,

$$\frac{\partial H}{\partial H_0} = E(z), \tag{B.1}$$

Derivatives of the function $\mathscr{E}(z) \equiv H^2(z)/H_0^2 = E^2(z)$ are found in all derivatives of both H and d_A and are worth defining separately:

$$\begin{aligned}
\frac{\partial \mathscr{E}(z)}{\partial \Omega_m} &= (1+z)^3 - f(z), \\
\frac{\partial \mathscr{E}(z)}{\partial \Omega_k} &= (1+z)^2 - f(z), \\
\frac{\partial \mathscr{E}(z)}{\partial w_0} &= (1-\Omega_m-\Omega_k)\frac{\partial f(z)}{\partial w_0}, \\
&= 3(1-\Omega_m-\Omega_k) f(z) \ln(1+z), \\
\\
\frac{\partial \mathscr{E}(z)}{\partial w_a} &= (1-\Omega_m-\Omega_k)\frac{\partial f(z)}{\partial w_a}, \\
&= 3(1-\Omega_m-\Omega_k) f(z) \left(\ln(1+z) - \frac{z}{1+z}\right).
\end{aligned} \tag{B.2}$$

For all the cosmological parameters we consider other than the Hubble parameter H_0, the derivatives with respect to the Hubble parameter can then be expressed as,

$$\frac{\partial H(z)}{\partial \theta_i} = \frac{H_0}{2E}\frac{\partial \mathscr{E}(z)}{\partial \theta_i}, \quad \theta_i \in (\Omega_m, \Omega_k, w_0, w_a). \tag{B.3}$$

B.1.2 Angular Diameter Distance

In a FLRW background, $d_A(z)$ is given by Eq (5.5). The function $\chi(z)$ is defined by Eq. (5.6). It is again useful to define first order derivatives of $\chi(z)$ with respect to the parameters, as these derivatives will occur many times in the Fisher derivatives of $d_A(z)$:

$$\frac{\partial \chi(z)}{\partial \theta_i} = -\int_0^z \frac{1}{2E^3(z')}\frac{\partial \mathscr{E}(z')}{\partial \theta_i} dz', \quad \theta_i \in (\Omega_m, \Omega_k, w_0, w_a)\,, \tag{B.4}$$

again where $\mathscr{E}(z) \equiv H^2(z)/H_0^2$. The Hubble parameter appears only in the pre-factor of the angular diameter distance, and hence:

$$\begin{aligned}
\frac{\partial d_A(z)}{\partial H_0} &= -\frac{1}{1+z}\frac{c}{H_0^2}\frac{1}{\sqrt{\Omega_k}}\sinh\left(\sqrt{\Omega_k}\chi(z)\right), \\
&= \left(-\frac{1}{H_0}\right) d_A(z).
\end{aligned} \tag{B.5}$$

The matter density Ω_m contributes solely to the $\chi(z)$ term, and hence can be expressed using Eqs. (B.4) and (B.2) as,

$$\frac{\partial d_A(z)}{\partial \Omega_m} = \frac{1}{1+z}\frac{c}{H_0}\cosh\left(\sqrt{\Omega_k}\chi(z)\right)\frac{\partial \chi(z)}{\partial \Omega_m}, \tag{B.6}$$

with the derivatives of the co-moving distance given by,

$$\frac{\partial \chi(z)}{\partial \theta_i} = -\frac{1}{2}\int_0^z \frac{1}{E^3(z')}\frac{\partial \mathscr{E}(z')}{\partial \theta_i}dz' \quad \text{for } \theta_i \in (\Omega_m, \Omega_k, w_0, w_a). \tag{B.7}$$

The curvature parameter is found both in the pre-factor and the sinh term of the angular diameter distance, hence:

$$\begin{aligned}\frac{\partial d_A(z)}{\partial \Omega_k} &= -\frac{1}{1+z}\frac{c}{H_0}\frac{1}{2\Omega_k^{3/2}}\sinh\left(\sqrt{\Omega_k}\chi(z)\right) \\ &\quad + \frac{1}{1+z}\frac{c}{H_0}\frac{1}{\sqrt{\Omega_k}}\cosh\left(\sqrt{\Omega_k}\chi(z)\right), \left[\frac{\chi(z)}{2\sqrt{\Omega_k}} + \sqrt{\Omega_k}\frac{\partial \chi(z)}{\partial \Omega_k}\right] \\ &= -\frac{1}{2\Omega_k}d_A(z) + \frac{1}{1+z}\frac{c}{H_0}\cosh\left(\sqrt{\Omega_k}\chi(z)\right)\left[\frac{\chi(z)}{2\Omega_k} + \frac{\partial \chi(z)}{\partial \Omega_k}\right].\end{aligned} \tag{B.8}$$

The Taylor series expansion of Eq (B.8) is used $\Omega_k \to 0$, namely:

$$\left.\frac{\partial d_A(z)}{\partial \Omega_k}\right|_{\Omega_k \to 0} = \frac{c}{H_0}\frac{1}{1+z}\left\{\frac{1}{6}\chi^3(z,0) + \frac{\partial \chi(z,0)}{\partial \Omega_k}\right\}, \tag{B.9}$$

where,

$$X(z,0) \equiv X(z)|_{\Omega_k \to 0}\,, \tag{B.10}$$

are the functions (for example $E(z)$, $\chi(z)$) assuming flatness. Using the definitions of Eqs. (B.4) and (B.2), the derivatives of the angular diameter distance are expressed similarly for $\theta_i \in (w_0, w_a)$ as:

$$\frac{\partial d_A(z)}{\partial \theta_i} = \frac{1}{1+z}\frac{c}{H_0}\cosh\left(\sqrt{\Omega_k}\chi(z)\right)\frac{\partial \chi(z)}{\partial \theta_i}\,. \tag{B.11}$$

Appendix C

Fisher4Cast - Getting Started

Currently the code is available for download at one of the following websites [113,120]. Save this '.zip' file into the directory you want to run the Fisher4Cast suite from.

C.1 The Graphical User Interface

The GUI can be started from the Matlab editor. The file **FM_GUI.m** must be opened from the directory, and once the file is opened (click on the file icon from within the Command-line interface to open it with an editor) press 'F5' to run the code. This will open up the GUI screen.

You can also launch the GUI from the command line by typing:

```
>>FM_GUI
```

The output data will not be saved into the workspace, but the 'Saving Features' button allows one to save the input and output from any particular run in text or LaTeX code.

C.1.1 The Basic Layout Explained

We describe the basic layout of the GUI, and illustrate the various actions with screenshots taken of a working GUI.

The GUI has three main sections. The section on the top left controls the input to the GUI. The bottom left panel controls the things one might like to use in the analysis and the parameters you are interested in plotting. In Figure (C.1) we show the initial GUI screen, highlighting the observable about to be used (here the Growth

function $G(z)$), and the cosmological parameters relevant to the analysis (which are the w_0 and w_a coefficients in the Chevallier-Polarski-Linder parameterisation of dark energy [8, 121] – see Eq. (5.1)). The specific cosmological example is described in detail in [2], which contains the set of analytical derivatives used in Fisher4Cast. The right-hand side of the GUI controls the plotting commands for the ellipse. The various actions used to control the output are described below.

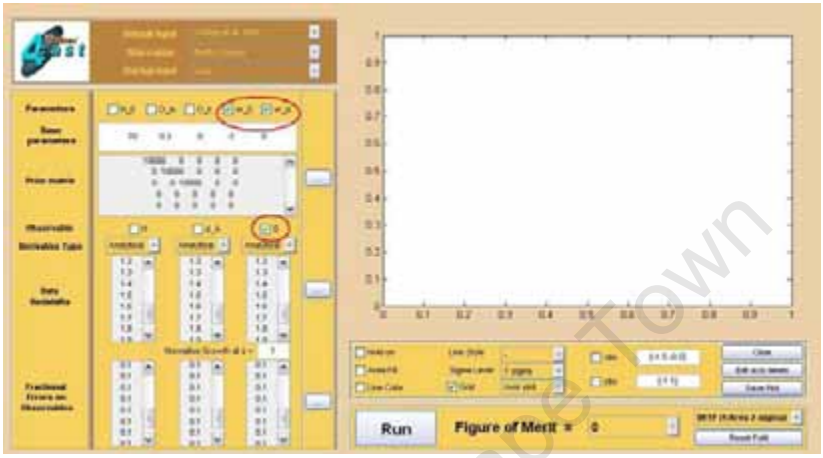


Figure C.1: The start-up screen of the Fisher4Cast GUI.

C.1.2 Changing the Input Structure

In order to compute Fisher ellipses for different input structures, one can either choose from a drop down list of default example structures contained within the distribution (as shown in Figure (C.2)) or one can generate a unique input structure. This file can then be loaded to the GUI, which must be given as '.m' file. You can obviously also just edit the input parameters in the GUI after the default input has been loaded or alternately you can edit the input file (eg **Cooray_et_al_2004.m**).

C.1.3 Floating Help

Floating help is provided with the Fisher4Cast GUI for most commands. The floating help is activated by moving the mouse pointer over the button or parameter on the GUI and leaving it there for a few seconds. This generates a screen prompt, which pops up and gives information about the function of the button or parameter in question. Figure (C.3) shows this help prompt for the 'Run' button on the GUI.

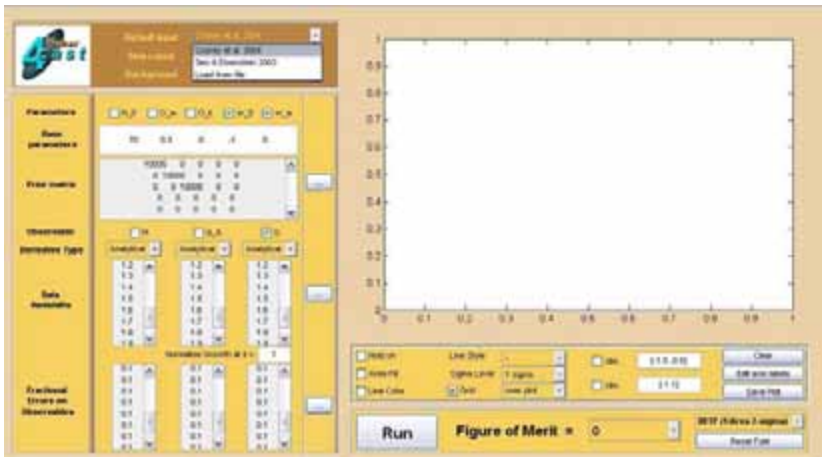


Figure C.2: Changing the default input structure from the drop-down menu.

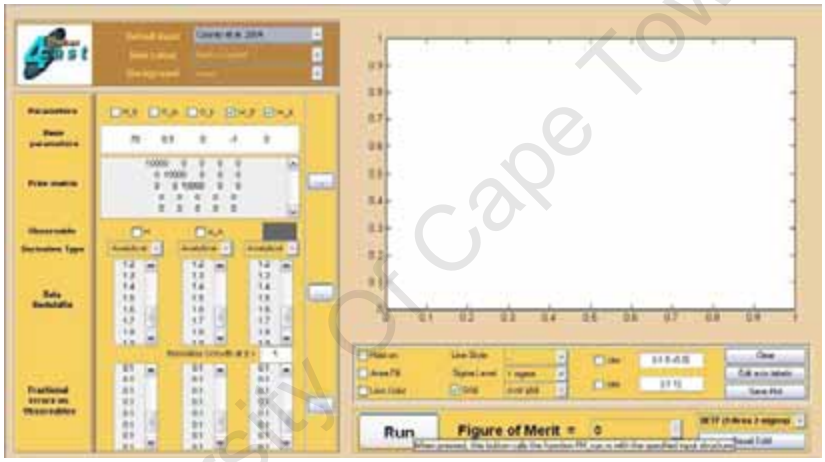


Figure C.3: The floating help for the 'Run' button.

C.1.4 Running Fisher4Cast

Once satisfied with the observables considered and the parameters of interest, pressing the 'Run' button will execute the code. A box will pop up that will state that the code is running, and an error ellipse will appear when the code has finished running. This is shown in Figure (C.4).

C.1.5 Errors in the Input

When the 'Run' button is pushed, the GUI first calls the **FM_errorchecker.m** function with the input supplied. This checks for the input files, checks that the data vector (e.g. redshifts at which one has measurements of the Hubble parameter) and

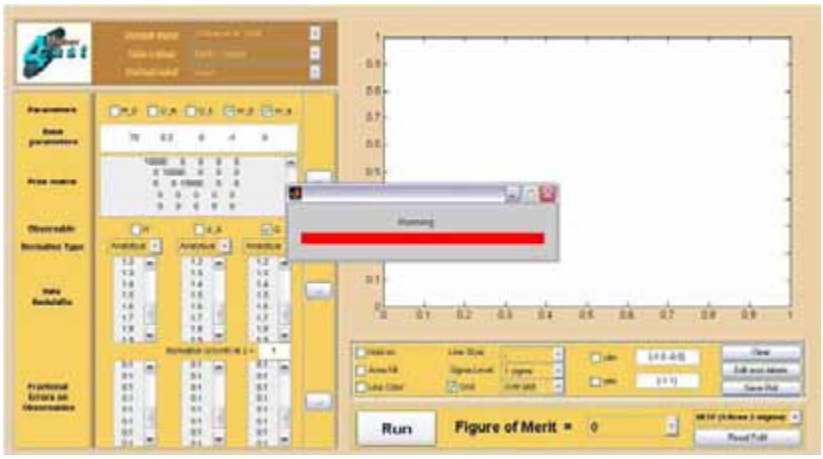


Figure C.4: Running the Fisher4Cast from the GUI.

error vectors (e.g. the fractional errors on the Hubble parameter, σ_H/H, at the redshifts above) are the same length and performs other consistency checks. Should any of these tests fail, an error box will appear explaining which errors to fix before calling the GUI again. A log file of these errors is created in the same directory the GUI is being run in, and is called 'log.mat'. Loading and reading this log file is described in Section C.3. Figure (C.5) shows the error dialogue box indicating that a single error has been found.

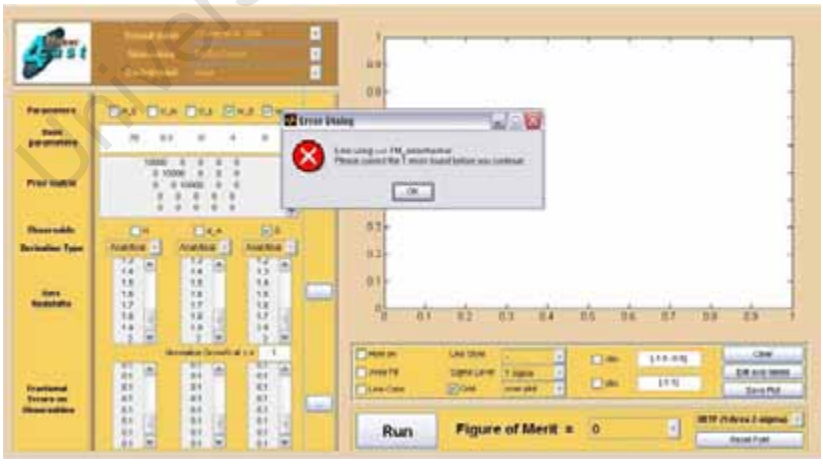


Figure C.5: The GUI error message box. The errors are detailed in 'log.mat'.

C.1.6 The Fisher Ellipse

Once the code is running smoothly, the resulting Fisher error ellipse is plotted. This is shown in Figure (C.6).

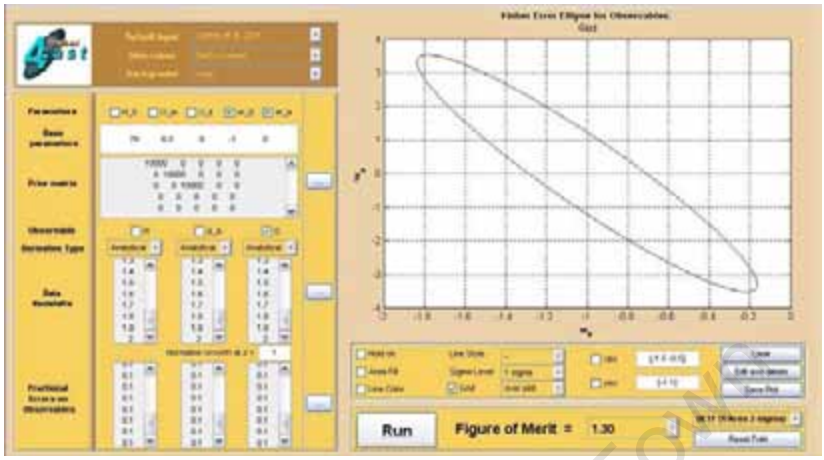


Figure C.6: The Fisher ellipse resulting from a run of Fisher4Cast.

C.1.7 Plotting more than one Ellipse

Should one want to superimpose more than one ellipse, click the 'Hold on' button. This works both for the line and the area (although the same line and area fill properties will be used for both ellipses – see the below item for discussion of changing the colour of the area fill). Figure (C.7) shows the resulting ellipse for two observables, $G(z), d_A(z)$.

C.1.8 Area fill

Clicking on this button yields a filled error ellipse. Once it is clicked a colour must be selected from the menu on the left pop-up box. Note that should more than one error ellipse be plotted later, this area fill box must be ticked and un-ticked again to change the colour, otherwise the same colour will be used for all filled ellipses. This box is shown in Figure (C.8).

C.1.9 Importing Data

The input data can also be imported from a file - either as the redshift vector (the data), the error vector or the matrix of prior information on the cosmological parameters. This can be done by clicking the relevant 'Browse' buttons on the GUI.

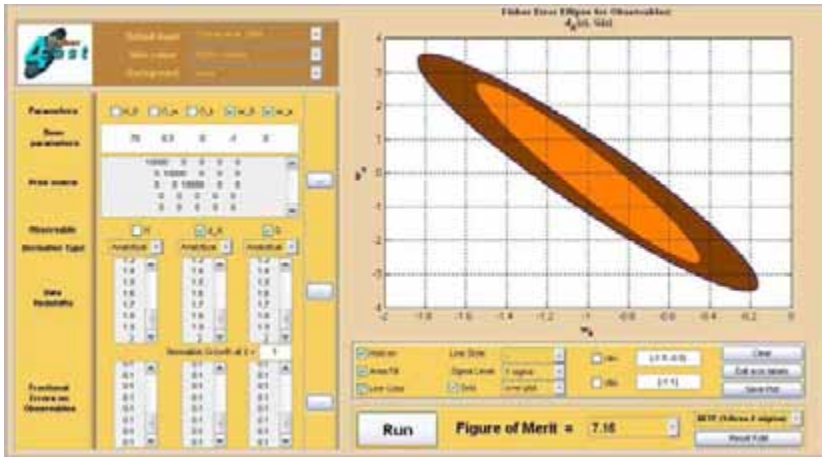


Figure C.7: Multiple ellipses are plotted in one figure.

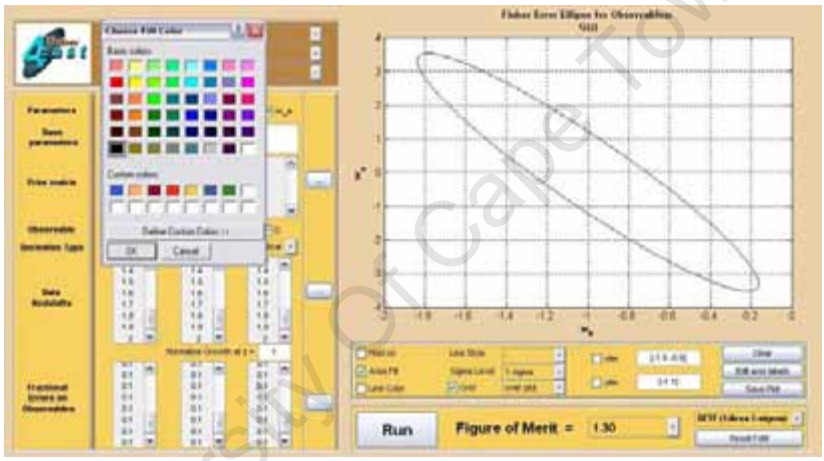


Figure C.8: The Area Fill option with colour selection.

This brings up a screen in which one can either load the data from file, or from the clipboard, in which case the data is cut and paste into the GUI fields. Figure (C.9) shows the screens for the loading of data from a file in the directory. In addition there is a check-box which specifies whether or not to use the prior matrix.

C.1.10 Multiple σ

It is possible to plot the ellipses for multiple confidence levels (i.e. $67\%, 95\%, 99\%$ specified by $1-, 2-,$ and $3-\sigma$ respectively). This is done via a drop-down menu on the right-hand side of the GUI, and is illustrated in Figure (C.10). It is worth noting that the 'Hold on' was used to generate this plot.

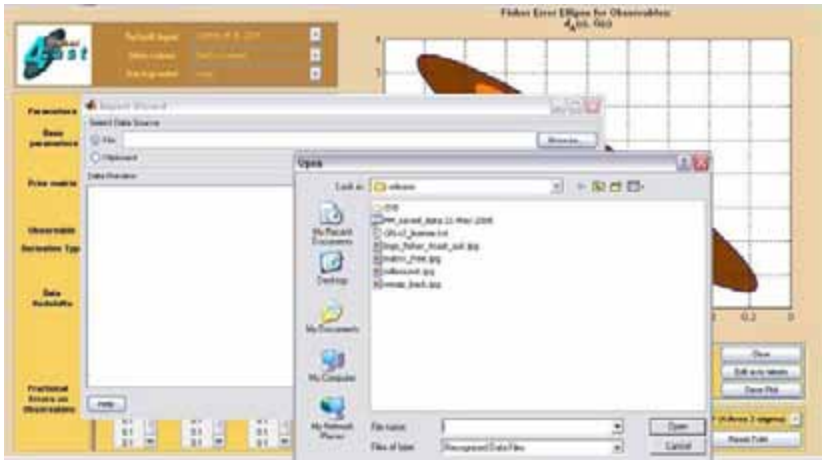


Figure C.9: Loading data into the GUI.

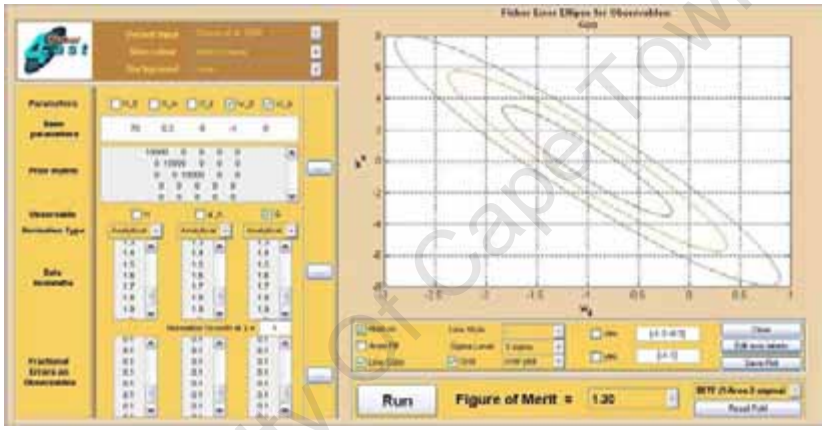


Figure C.10: Ellipses with many σ.

C.1.11 Different Definitions of the Figure of Merit

Fisher4Cast provides various Figures of Merit in order to compare different surveys. These are defined in detail in Section 6.2.3. Figure (C.15) illustrates how these various FoMs are accessible in the GUI.

C.1.12 Controlling Output

The buttons on the right-hand side of the GUI all control the output specifications of the ellipse, such as the limits of the x and y axis, the line style and colour of the ellipse, and whether or not to have a grid on (over or under) the data. This is designed for maximum flexibility in representing the ellipses in a unique and distinguishable way. The axis labels can also be modified using the 'Edit Axis' button.

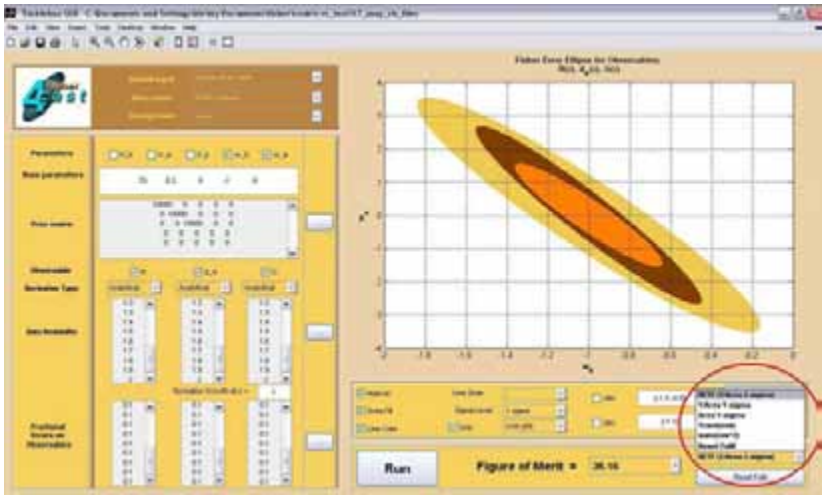


Figure C.11: Various Figures of Merit σ are calculated in Fisher4Cast.

C.1.13 Saving the Plot

Once satisfied with the ellipses plotted, figures are saved by clicking on the 'Saving Features' menu and selecting the 'Save Plot' option. This will bring up a window to save the figure to a particular directory, in a selected file format ('.eps', '.fig', '.png', '.pdf' etc.), as illustrated in Figure (C.12). No legend information (for example which colours correspond to which ellipses) will be saved. If legend entries are required, the figure must be saved as a '.fig' file, opened with Matlab, with the required legend entries added later. In addition to the simple saving of the figure, the reporting

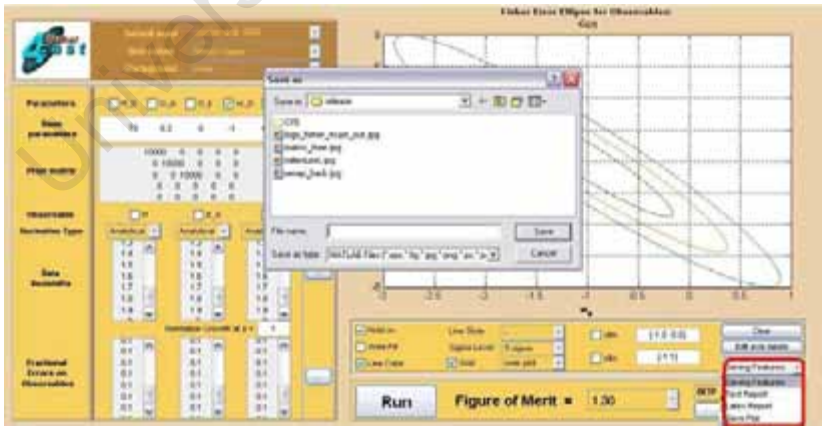


Figure C.12: Saving the figure of your Fisher4Cast ellipse.

features of the Fisher4Cast code are accessed through this 'Saving Features' menu. In short, they provide the opportunity to save a report of the inputs and the resulting

outputs and Fisher matrices from the survey, in either ASCII text or LaTeX format. When clicking the 'Text report' feature the user is prompted to save the resulting '.txt' file. Similarly if one chooses the LaTeX report, both a '.tex' report and an Encapsulated Postscript File will be generated, and the user is prompted to save both the figure and the LaTeX document. These reporting features are discussed in more detail in Section 6.3.1.

C.1.14 Skins

The GUI is available in a variety of skins and backgrounds. These can be chosen from a drop-down list (consisting of both colour schemes and background images [122, 123]); additional background images can be loaded by the user. This is shown in Figure (C.13).

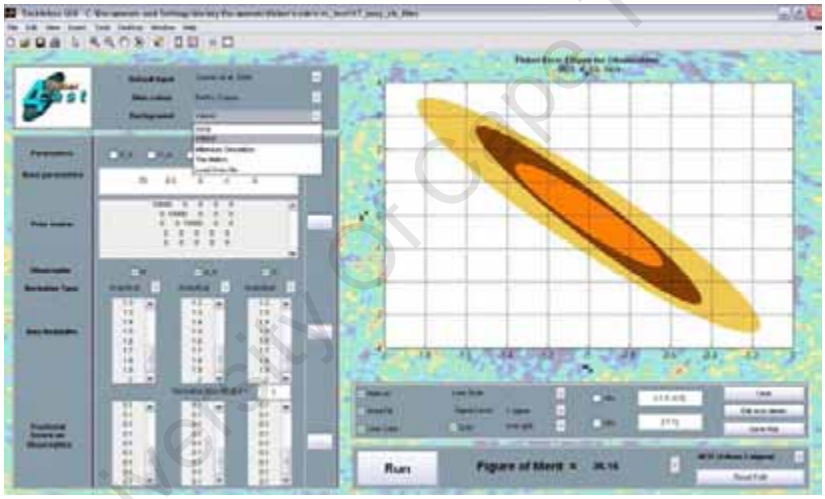


Figure C.13: Changing the skins and background images.

C.1.15 Fisher4Cast Menu

A Fisher4Cast menu is defined in the top left-hand corner of the Fisher4Cast GUI. From this drop-down menu one can access the **Readme** file of the code suite, the **Users' manual** and **Quickstart Guide** for easy reference, and the version history of the code. The BSD licence [124] for the Fisher4Cast suite is also available from the drop-down list. This list is illustrated in Figure (C.15). In addition to the Fisher4Cast menu, there is a menu with information on the extensions available

for Fisher4Cast. In future releases of the code this menu will select the extensions themselves, at present it provides the **Readme** for the modules.

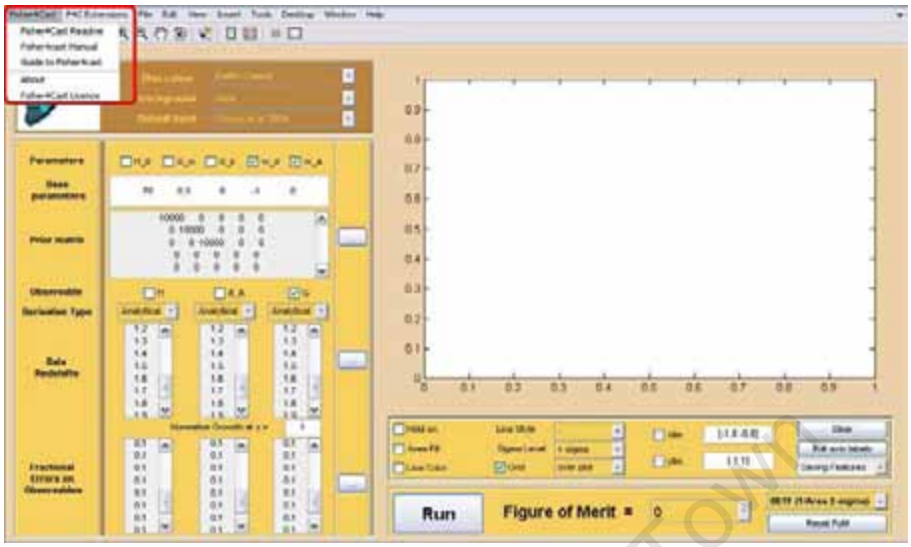


Figure C.14: The Fisher4Cast drop-down menu with information on the code and version.

C.1.16 Interactive Plotting

Interactive 'point-and-click' plotting is available in Fisher4Cast Version 1.2, available by selecting the 'Activate Interactive Plotting' option from the 'Fisher4Cast Extension' menu. Once selected the user interactively sets the values for parameters being plotted by clicking on the plotting area of the GUI. The arrow is activated for the first click on the plot area, the next click will run the code and produce the appropriate ellipse. The values selected will be displayed in the parameter input section of the GUI, the same as they would have been if manually entered. Care should be taken not to step to very unphysical values of the parameters, e.g. very positive values of w_0.

C.2 The Command Line

C.2.1 Running the Code

Open your version of Matlab and change the working directory to be the same as where you saved Fisher4Cast in. To run the code from the command line with one of the standard test input structures supplied, type:

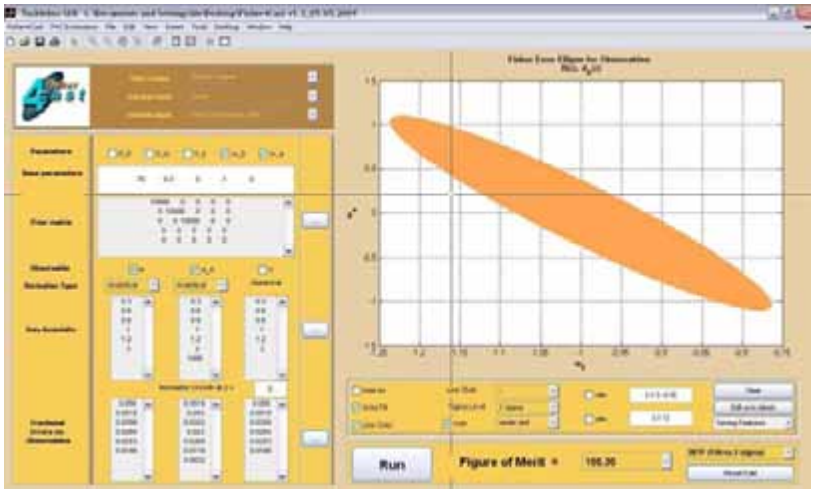


Figure C.15: The Interactive Plotting Feature in the Fisher4Cast code. This allows you to click on the parameter plane to change the assumed fiducial model.

```
>>output = FM_run(Cooray_et_al_2004)
```

This will call the code using the pre-supplied test input data (**Cooray_et_al_2004**) and then generate an error ellipse plot for the parameters and observables supplied in the chosen input. All the relevant generated output is written to the output structure. You can see the range of outputs to access by typing:

```
>>output
```

and then examine each output individually by specifying it exactly. For example:

```
>>output.marginalised_matrix
```

will access the marginalised_matrix field in the output structure. It is worth noting that each '.' denotes another sub-level in the input structure.

Example input files are supplied as a template for generating new input files with your own customised parameters and values. All fields specified in the example inputs must be specified in any user-defined example input. These are outlined in Section 6.2.1.

The code can also be run from the Matlab editor. Once the code is opened (open it from inside the Matlab window), pressing 'F5' will run the code. Note that if the code is run from the Editor it will call the default input structure, which is the **Cooray_et_al_2004.m** file. This is an example file containing input data from the paper by Cooray *et al.* [115]. This output can be directly compared to that of Figure 1 of that paper. If your output compares correctly, you have a working installation of the code. Another input available is **Seo_Eisenstein_2003.m** [14].

C.3 FM_errorchecker

The error-checker function acts 'behind-the-scenes' to check that the input structure and all the required variables are correct before executing the code. It can be run directly by using the command:

```
>>FM_errorcheck(FM_initialise)
```

where **FM_initialise** is the specific function to initialise the input structure. The error checker validates, among other things, that all the derivative functions (whether analytical or numerical derivatives are going to be implemented) do in fact exist and that the data and corresponding variances vectors are the same length. This error checker is continually being updated to facilitate ease of use of the code. All error and checking messages are displayed to the screen and are also saved in the Matlab file 'log.mat' which can be loaded and examined at a later stage by invoking the following command:

```
>>load(log.mat)
```

C.4 Obtaining Baryon Acoustic Oscillation Errors from Survey Parameters

Two modules are included that calculate errors on the Hubble parameter and angular diameter distance in BAO surveys characterised in input structures of survey parameters. The provided codes are extensions to Fisher4Cast, and should be placed in the same folder as the main code suite so that they can access the required elements of the Fisher4Cast suite.

The first of these extensions, **EXT_FF_Blake_etal2005** uses the fitting formulae of Blake *et al.* [13] to calculate the errors on the Hubble parameter and angular diameter distance given certain survey specifications, such as the survey area and the redshifts used for the measurements of H and d_A. These are either given as central redshift bins or as the edges of redshift bins. The galaxy number density is also required, and is expected in units of $10^{-3}\mathrm{Mpc}^{-3}h^3$.

All files associated with this module have the same prefix to identify them as external modules for the fitting formula of Blake *et. al* [13]. The fitting formulae contain coefficients specific to either photometric or spectroscopic surveys, and hence it must be specified which survey one is considering. The input parameters to this module are supplied in an input structure, which is explained in the **EXT_FF_Blake_etal2005_Readme.txt** file. They are:

- Input_survey.base_parameters: The fiducial values of the cosmological parameters are specified here, as $(H_0, \Omega_m, \Omega_K, w_0, w_a)$, in the same way as the input parameters are defined in the main Fisher4Cast code.

- Input_survey.surv_type: This specifies either a photometric or spectroscopic survey by either setting it to 'spec' or 'phot'. This defines which of the sets of coefficients to use in the fitting formulae.

- Input_survey.z_type: This field indicates that the redshift data being used are either given at the 'edge' of the bins or as the 'central' redshifts. If Input_survey.z_type='central' then an additional input for width of the redshift bins, dz, is required.

- Input_survey.area: This field specifies the area of the survey in units of 1000 square degrees.

- Input_survey.n: This gives the number density of galaxies and is measured in units of $10^{-3}\mathrm{h}^3\mathrm{Mpc}^{-3}$

- Input_survey.vecH and Input_survey.vecDA: These entries give the redshift vectors of the survey for the Hubble parameter $H(z)$ and angular diameter distance. $d_A(z)$

- Input_survey.biasH and Input_survey.biasDA : These entries specify the bias for the Hubble parameter and angular diameter distance. These two vectors should have the same length as that of 'Input_survey.vecH' and 'Input_survey.vecDA' respectively.

An example of a default input structure supplied in the file **EXT_FF_Blake_etal2005_Input.m**. The command used to run the Blake *et al.* [13] fitting formula is given as:

```
>>[z_H_central,z_DA_central,vol_H, vol_DA,sigH, sigDA] = ...
>>EXT_FF_Blake_etal2005_Main(Input_survey)
```

Should no input structure be specified when calling the code, the default input (**EXT_FF_Blake_etal2005_Input.m**) is assumed. The module comprises of various smaller modules to compute the oscillation scales in the radial and transverse direction, and hence the errors on the angular diameter distance and Hubble parameter. Figure (C.16) outlines the procedures in the module. The module outputs the following parameters:

- z_H_central: A vector of the central redshifts of the $H(z)$ data bins, related to the Input_survey.vecH fields specified in the input structure.
- z_DA_central: A vector of the central redshifts of the $d_A(z)$ data bins, related to the Input_survey.vecH fields specified in the input structure.
- vol_H: A vector of the volume of the redshift bins for the Hubble parameter.
- vol_DA: A vector of the volume of the redshift bins for the angular diameter distance.
- sigH: A vector of the *fractional* error on the Hubble parameter: σ_H/H. For percentage errors, multiply this by 100.
- sigH: A vector of the *fractional* error on the angular diameter distance: σ_{d_A}/d_A. For percentage errors, multiply this by 100.

The module **EXT_FF_SeoEisenstein2007** also computes the errors on the Hubble parameter and angular diameter distance using the prescription set out in [117] and the sound horizon scale as given in [118]. This module contains a wrapper to call the Matlab version of the **C** code of Seo & Eisenstein [119]. In this module the code does not need to be in the same directory as the Fisher4Cast suite, and runs completely independently of Fisher4Cast.

The module also takes an input structure (a default input structure with all fields specified is given in **EXT_FF_SeoEisenstein2007_Input.m**) with the following parameters defined:

- **Input_survey.number_density**: The galaxy number density in units of h^3 Mpc^{-3}
- **nput_survey.wmap**: A flag (1 or 3) specifying which WMAP power spectrum to use.
- **Input_survey.sigma8**: The value of σ_8, the amplitude of linear clustering on a scale of 8 Mpc.
- **Input_survey.Sigma_z** The line of sight real mean squared comoving distance error due to redshift uncertainties.
- **Input_survey.beta**: The redshift distortion parameter is entered.
- **Input_survey.volume**: The survey volume in units of h^{-3} Gpc^3.

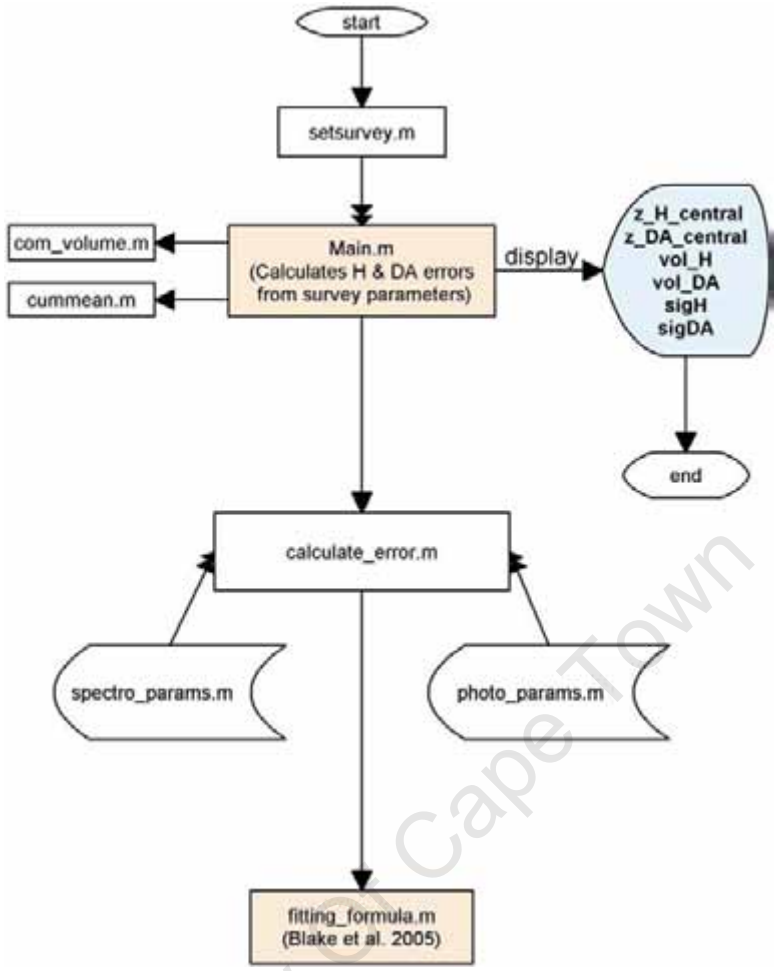


Figure C.16: The flow of inputs and functions within the Blake *et al.* [13] extension module to Fisher4Cast.

Sigma_perp (the transverse rms Lagrangian displacement) and Sigma_par (the radial displacement) are calculated and saved to the input structure. The module is comprised of smaller functions; the flowchart of the module is shown in Figure (C.17).

As in the case of the Blake *et al.* module, the Seo and Eisenstein module code can be called from the command line using:

```
>>[Drms,Hrms,r,Rrms] = EXT_FF_SeoEisenstein2007_Main(Input_survey)
```

The outputs of the code are the root mean square error on D/s and Hs, where s is the oscillation scale. These are both given as fractional error, for percentage error multiplied by 100. In addition the correlation coefficient between D and H is given (r), and the diagonal entry in the covariance matrix between D and H.

To be sure that the extensions have access to the functions contained in Fisher4Cast we need to be sure that the extensions are either placed directly in the same folder or

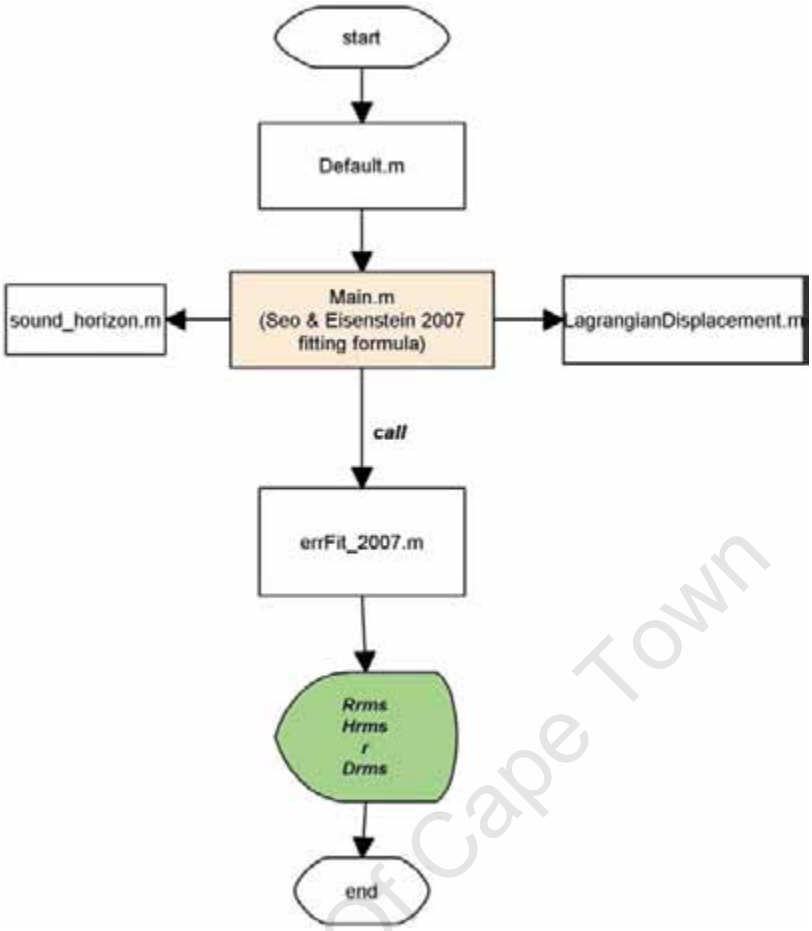


Figure C.17: Flowchart of the code in the Seo and Eisenstein BAO extension module to Fisher4Cast.

the path is specified to both Fisher4Cast and the extensions. One can use the path command to do this:

```
>>path(path,'/path-to-folder/Fisher4Cast-v2.0')
>>path(path,'/path-to-folder/EXT_FF_Blake_etal2005')
```

where the 'path-to-folder' is the path specifying the directory where the extension or Fisher4Cast code is kept on your local computer. Similarly, to run the Seo and Eisenstein [117] module, you will need to ensure that the respective extension is either in the same directory or the path is specified:

```
>>path(path,'/path-to-folder/Fisher4Cast-v2.0')
>>path(path,'/path-to-folder/EXT_FF_SeoEisenstein2007')
```

All files associated with this module have the same prefix to identify them as external modules for the fitting formula of Blake *et. al* [13]. The fitting formulae contain coefficients specific to either photometric or spectroscopic surveys, and

hence it must be specified which survey one is considering. The input parameters to this module are supplied in an input structure, which is explained in the **EXT_FF_Blake_etal2005_Readme.txt** file. They are:

- Input_survey.base_parameters: The fiducial values of the cosmological parameters are specified here, as $(H_0, \Omega_m, \Omega_K, w_0, w_a)$, in the same way as the input parameters are defined in the main Fisher4Cast code.

- Input_survey.surv_type: This specifies either a photometric or spectroscopic survey by either setting it to 'spec' or 'phot'. This defines which of the sets of coefficients to use in the fitting formulae.

- Input_survey.z_type: This field indicates that the redshift data being used are either given at the 'edge' of the bins or as the 'central' redshifts. If Input_survey.z_type='central' then an additional input for width of the redshift bins, dz, is required.

- Input_survey.area: This field specifies the area of the survey in units of 1000 square degrees.

- Input_survey.n: This gives the number density of galaxies and is measured in units of $10^{-3}\mathrm{h}^3\mathrm{Mpc}^{-3}$

- Input_survey.vecH and Input_survey.vecDA: These entries give the redshift vectors of the survey for the Hubble parameter $H(z)$ and angular diameter distance. $d_A(z)$

- Input_survey.biasH and Input_survey.biasDA : These entries specify the bias for the Hubble parameter and angular diameter distance. These two vectors should have the same length as that of 'Input_survey.vecH' and 'Input_survey.vecDA' respectively.

Bibliography

[1] B. A. Bassett, M. Brownstone, A. Cardoso, et al. *Is the dynamics of scaling dark energy detectable?* JCAP, 7:7–+ (2008). arXiv:0709.0526.

[2] B. A. Bassett, Y. Fantaye, R. Hlozek, et al. *Fisher Matrix Preloaded – Fisher4Cast*. ArXiv e-prints (2009). arXiv:0906.0993.

[3] B. A. Bassett, Y. Fantaye, R. Hlozek, et al. *Fisher4Cast Users' Manual*. ArXiv e-prints (2009). arXiv:0906.0974.

[4] A. Courty, A.-I. Henry, N. Goubet, et al. *Large triangular single crystals formed by mild annealing of self-organized silver nanocrystals*. Nat Mater, 6(11):900–907 (2007).

[5] W. Hu and . White, M. J. *The cosmic symphony*. Sci. Am., 290N2:32–37 (2004).

[6] M. Kowalski, D. Rubin, G. Aldering, et al. *Improved Cosmological Constraints from New, Old, and Combined Supernova Data Sets*. Astrophys. J., 686:749–778 (2008). arXiv:0804.4142.

[7] S. Burles, K. M. Nollett, and M. S. Turner. *Big-Bang Nucleosynthesis: Linking Inner Space and Outer Space*. ArXiv e-prints (1999). arXiv:astro-ph/9903300.

[8] M. Chevallier and D. Polarski. *Accelerating Universes with Scaling Dark Matter*. Int. J. Mod. Phys. D, 10:213–223 (2001).

[9] E. V. Linder. *Exploring the Expansion History of the Universe*. Phys. Rev. Lett., 90(9):091301–+ (2003). arXiv:astro-ph/0208512.

[10] E. F. Bunn and M. White. *The 4 Year COBE Normalization and Large-Scale Structure*. Astrophys. J., 480:6–+ (1997). arXiv:astro-ph/9607060.

[11] M. R. Nolta, J. Dunkley, R. S. Hill, et al. *Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Angular Power Spectra*. ApJS, 180:296–305 (2009). arXiv:0803.0593.

[12] J. Dunkley, E. Komatsu, M. R. Nolta, et al. *Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Likelihoods and Parameters from the WMAP data*. ApJS, 180:306–376 (2009). arXiv:0803.0586.

[13] C. Blake, D. Parkinson, B. Bassett, et al. *Universal fitting formulae for baryon oscillation surveys*. MNRAS, 365:255–264 (2006). arXiv:astro-ph/0510239.

[14] H.-J. Seo and D. J. Eisenstein. *Probing Dark Energy with Baryonic Acoustic Oscillations from Future Large Galaxy Redshift Surveys*. Astrophys. J., 598:720–740 (2003). arXiv:astro-ph/0307460.

[15] D. Schlegel, M. White, and D. Eisenstein. *The Baryon Oscillation Spectroscopic Survey: Precision measurements of the absolute cosmic distance scale*. ArXiv e-prints (2009). arXiv:0902.4680.

[16] K. Glazebrook, C. Blake, W. Couch, et al. *The WiggleZ project: AAOmega and Dark Energy*. ArXiv e-prints (2007). arXiv:astro-ph/0701876.

[17] A. Papoulis. *Probability and statistics*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1990). ISBN 0137116985.

[18] A. Leon-Garcia. *Probability, Statistics, and Random Processes for Electrical Engineering*. Prentice Hall (2007). ISBN 131471228.

[19] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, 2 edition (2001). ISBN 0534243126.

[20] S. Dodelson. *Modern Cosmology*. Academic Press, Amsterdam (2003). ISBN 0122191412.

[21] M. Tegmark, A. N. Taylor, and A. F. Heavens. *Karhunen-Loeve Eigenvalue Problems in Cosmology: How Should We Tackle Large Data Sets?* Astrophys. J., 480:22–+ (1997). arXiv:astro-ph/9603021.

[22] T. Matsubara. *Correlation Function in Deep Redshift Space as a Cosmological Probe.* Astrophys. J., 615:573–585 (2004). `arXiv:astro-ph/0408349`.

[23] W. H. Press. *Numerical recipes in C++ : the art of scientific computing.* Cambridge University Press (2002). ISBN 0521750334.

[24] B. A. Bassett, D. Parkinson, and R. C. Nichol. *Designer Cosmology.* Astrophys. J., 626:L1–L4 (2005). `arXiv:astro-ph/0409266`.

[25] J. Li, W. keng Liao, A. Choudhary, et al. *I/O Analysis and Optimization for an AMR Cosmology Application.* Cluster Computing, IEEE International Conference on, 0:119 (2002).

[26] E. C. Elson, B. A. Bassett, K. van der Heyden, et al. *A new approach to the optimal target selection problem.* A&A, 464:1167–1175 (2007). `arXiv:astro-ph/0602338`.

[27] C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, Oxford, UK (1996). ISBN 0198538499.

[28] M. F. Møller. *A scaled conjugate gradient algorithm for fast supervised learning.* Neural Netw., 6(4):525–533 (1993).

[29] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16).* Soc. for Industrial & Applied Math (1996). ISBN 0898713641.

[30] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford University Press, Oxford, UK (1996). ISBN 0195099710.

[31] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Basic Algorithms and Operators.* IOP Publishing Ltd., Bristol, UK, UK (1999). ISBN 0750306645.

[32] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Advanced Algorithms and Operators.* IOP Publishing Ltd., Bristol, UK, UK (1999). ISBN 0750306653.

[33] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989). ISBN 0201157675.

[34] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Bradford Book. The MIT Press (2004). ISBN 0262042193.

[35] M. Dorigo and C. Blum. *Ant colony optimization theory: a survey*. Theor. Comput. Sci., 344(2-3):243–278 (2005).

[36] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. *Optimization by simulated annealing*. Science, 220:671–680 (1983).

[37] J. Kennedy and R. C. Eberhart. *Particle swarm optimization*. Neural Networks, Proceedings of the IEEE International Conference on, 4:1942–1948 (1995).

[38] R. Eberhart and J. Kennedy. *A new optimizer using particle swarm theory*. Micro Machine and Human Science, Proceedings of the Sixth International Symposium on, pages 39–43 (1995).

[39] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, et al. *Equation of state calculations by fast computing machines*. J. J. Chem. Phys., 21(6):1087–1092 (1953).

[40] W. K. Hastings. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*. Biometrika, 57(1):97–109 (1970).

[41] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer–Verlag (1993).

[42] S. Weinberg. *Cosmology*. Oxford University Press, USA (2008). ISBN 0198526822.

[43] C. W. Misner, J. A. Wheeler, and K. Thorne. *Gravitation (Physics Series)*. W. H. Freeman (1973). ISBN 0716703440.

[44] R. M. Wald. *General Relativity*. University Of Chicago Press (1984). ISBN 0226870332.

[45] C. Clarkson, B. Bassett, and T. H.-C. Lu. *A general test of the Copernican Principle*. Phys. Rev. Lett., 101:011301 (2008). `arXiv:0712.3457`.

[46] C. Zunckel and C. Clarkson. *Consistency Tests for the Cosmological Constant*. Phys. Rev. Lett., 101(18):181301 (2008).

[47] P. Astier, J. Guy, N. Regnault, et al. *The Supernova Legacy Survey: measurement of Ω_M, Ω and w from the first year data set.* A&A, 447:31–48 (2006). arXiv:astro-ph/0510447.

[48] W. J. Percival, S. Cole, D. J. Eisenstein, et al. *Measuring the Baryon Acoustic Oscillation scale using the Sloan Digital Sky Survey and 2dF Galaxy Redshift Survey.* MNRAS, 381:1053–1066 (2007). arXiv:0705.3323.

[49] E. Rozo, R. H. Wechsler, E. S. Rykoff, et al. *Cosmological Constraints from the SDSS maxBCG Cluster Catalog.* ArXiv e-prints (2009). arXiv:0902.3702.

[50] D. W. Hogg. *Distance measures in cosmology.* ArXiv e-prints (1999). arXiv:astro-ph/9905116.

[51] A. A. Penzias and R. W. Wilson. *A Measurement of Excess Antenna Temperature at 4080 Mc/s.* Astrophys. J., 142:419–421 (1965).

[52] G. F. Smoot, C. L. Bennett, A. Kogut, et al. *Preliminary results from the COBE differential microwave radiometers - Large angular scale isotropy of the cosmic microwave background.* Astrophys. J., 371:L1–L5 (1991).

[53] D. N. Spergel, R. Bean, O. Doré, et al. *Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology.* ApJS, 170:377–408 (2007). arXiv:astro-ph/0603449.

[54] G. Hinshaw, J. L. Weiland, R. S. Hill, et al. *Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Data Processing, Sky Maps, and Basic Results.* ApJS, 180:225–245 (2009). arXiv:0803.0732.

[55] P. J. E. Peebles and J. T. Yu. *Primeval Adiabatic Perturbation in an Expanding Universe.* Astrophys. J., 162:815–+ (1970).

[56] A. H. Guth. *The Inflationary Universe: A Possible Solution to the Horizon and Flatness Problems.* Phys. Rev. D, 23:347–356 (1981).

[57] R. K. Sachs and A. M. Wolfe. *Perturbations of a Cosmological Model and Angular Variations of the Microwave Background.* Astrophys. J., 147:73–+ (1967).

[58] J. Hwang, T. Padmanabhan, O. Lahav, et al. *1/3 factor in the CMB Sachs-Wolfe effect.* Phys. Rev. D, 65(4):043005–+ (2002). arXiv:astro-ph/0107307.

[59] M. Birkinshaw. *The Sunyaev-Zel'dovich effect*. Phys. Rep., 310:97–195 (1999). arXiv:astro-ph/9808050.

[60] A. Lewis, A. Challinor, and A. Lasenby. *Efficient Computation of CMB anisotropies in closed FRW models*. Astrophys. J., 538:473–476 (2000). arXiv:astro-ph/9911177.

[61] U. Seljak and M. Zaldarriaga. *A Line-of-Sight Integration Approach to Cosmic Microwave Background Anisotropies*. Astrophys. J., 469:437–+ (1996). arXiv:astro-ph/9603033.

[62] M. Zaldarriaga and U. Seljak. *CMBFAST for Spatially Closed Universes*. ApJS, 129:431–434 (2000). arXiv:astro-ph/9911219.

[63] A. G. Riess, A. V. Filippenko, P. Challis, et al. *Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant*. AJ, 116:1009–1038 (1998). arXiv:astro-ph/9805201.

[64] S. Perlmutter, G. Aldering, G. Goldhaber, et al. *Measurements of Omega and Lambda from 42 High-Redshift Supernovae*. Astrophys. J., 517:565–586 (1999). arXiv:astro-ph/9812133.

[65] A. G. Riess, P. E. Nugent, R. L. Gilliland, et al. *The Farthest Known Supernova: Support for an Accelerating Universe and a Glimpse of the Epoch of Deceleration*. Astrophys. J., 560:49–71 (2001). arXiv:astro-ph/0104455.

[66] A. Albrecht, G. Bernstein, R. Cahn, et al. *Report of the Dark Energy Task Force*. ArXiv e-prints (2006). arXiv:astro-ph/0609591.

[67] E. J. Copeland, M. Sami, and S. Tsujikawa. *Dynamics of Dark Energy*. International Journal of Modern Physics D, 15:1753–1935 (2006). arXiv:hep-th/0603057.

[68] R. R. Caldwell, R. Dave, and P. J. Steinhardt. *Cosmological Imprint of an Energy Component with General Equation of State*. Phys. Rev. Lett., 80:1582–1585 (1998). arXiv:astro-ph/9708069.

[69] P. G. Ferreira and M. Joyce. *Cosmology with a primordial scaling field*. Phys. Rev. D, 58(2):023503–+ (1998).

[70] E. J. Copeland, A. R. Liddle, and D. Wands. *Exponential potentials and cosmological scaling solutions.* Phys. Rev. D, 57:4686–4690 (1998). arXiv:gr-qc/9711068.

[71] R. Bean, S. H. Hansen, and A. Melchiorri. *Early-universe constraints on dark energy.* Phys. Rev. D, 64(10):103508–+ (2001).

[72] T. Barreiro, E. J. Copeland, and N. J. Nunes. *Quintessence arising from exponential potentials.* Phys. Rev. D, 61(12):127301–+ (2000).

[73] R. A. Alpher, G. Gamow, and R. Herman. *Thermal Cosmic Radiation and the Formation of Protogalaxies.* Proceedings of the National Academy of Science, 58:2179–2186 (1967).

[74] E. M. Burbidge, G. R. Burbidge, W. A. Fowler, et al. *Synthesis of the Elements in Stars.* Reviews of Modern Physics, 29:547–650 (1957).

[75] E. W. Kolb and M. S. Turner. *The early universe.* Addison-Wesley Longman Publishing Co., Inc. (1990).

[76] M. Doran, K. Karwan, and C. Wetterich. *Observational constraints on the dark energy density evolution.* JCAP, 11:7–+ (2005).

[77] J. Blomme, J. Debosscher, J. De Ridder, et al. *Automated classification of variable stars in the asteroseismology program of the Kepler space mission.* ArXiv e-prints (2010). arXiv:1001.0507.

[78] A. Prsa, E. F. Guinan, E. J. Devinney, et al. *Fully Automated Approaches to Analyze Large-Scale Astronomy Survey Data.* In *AGB Stars and Related Phenomenastro2010: The Astronomy and Astrophysics Decadal Survey*, volume 2010 of *Astronomy*, pages 25–+ (2009).

[79] J. S. Liu, F. Liang, and W. H. Wong. *The Multiple-Try Method and Local Optimization in Metropolis Sampling.* Journal of the American Statistical Association, (95):121–134 (2000).

[80] D. Frenkel and B. Smit. *Understanding Molecular Simulations: From Algorithms to Applications.* Academic Press (2002). ISBN 0122673514.

[81] B. A. Reid, W. J. Percival, D. J. Eisenstein, et al. *Cosmological Constraints from the Clustering of the Sloan Digital Sky Survey DR7 Luminous Red Galaxies.* ArXiv e-prints (2009). arXiv:0907.1659.

[82] E. J. Copeland, M. Sami, and S. Tsujikawa. *Dynamics of dark energy.* Int. J. Mod. Phys. D, 15(1753) (2006).

[83] P. S. Corasaniti, M. Kunz, D. Parkinson, et al. *The foundations of observing dark energy dynamics with the Wilkinson Microwave Anisotropy Probe.* Phys. Rev. D, 70:083006 (2004).

[84] B. A. Bassett, R. C. Nichol, D. J. Eisenstein, et al. A&G, 46:5.26 (2005).

[85] V. Sahni and L. Wang. *New cosmological model of quintessence and dark matter.* Phys. Rev. D, 62(10):103517–+ (2000).

[86] P. S. Corasaniti, M. Kunz, D. Parkinson, et al. *Foundations of observing dark energy dynamics with the Wilkinson Microwave Anisotropy Probe.* Phys. Rev. D, 70(8):083006–+ (2004). arXiv:astro-ph/0406608.

[87] D. Parkinson, C. Blake, M. Kunz, et al. *Optimizing baryon acoustic oscillation surveys - I. Testing the concordance ΛCDM cosmology.* MNRAS, 377:185–197 (2007).

[88] P. Zhang. *Self calibration of galaxy bias in spectroscopic redshift surveys of baryon acoustic oscillations.* ArXiv e-prints (2008). arXiv:0802.2416.

[89] W. J. Percival and M. White. *Testing cosmological structure formation using redshift-space distortions.* MNRAS, 393:297–308 (2009). arXiv:0808.0003.

[90] L. Amendola, C. Quercellini, and E. Giallongo. *Constraints on perfect fluid and scalar field dark energy models from future redshift surveys.* MNRAS, 357:429–439 (2005). arXiv:astro-ph/0404599.

[91] C. Di Porto and L. Amendola. *Observational constraints on the linear fluctuation growth rate.* ArXiv e-prints, 707 (2007). arXiv:0707.2686.

[92] A. Loeb and S. Wyithe. *Precise Measurement of the Cosmological Power Spectrum With a Dedicated 21cm Survey After Reionization.* ArXiv e-prints, 801 (2008). arXiv:0801.1677.

[93] Y. Wang. *Differentiating dark energy and modified gravity with galaxy redshift surveys.* ArXiv e-prints, 710 (2007). arXiv:0710.3885.

[94] E. V. Linder. *Cosmic growth history and expansion history.* Phys. Rev. D, 72(4):043529–+ (2005). arXiv:astro-ph/0507263.

[95] E. V. Linder. *Extending the gravitational growth framework.* Phys. Rev. D, 79(6):063519–+ (2009). arXiv:0901.0918.

[96] H. Zhan, L. Knox, and J. A. Tyson. *Distance, Growth Factor, and Dark Energy Constraints from Photometric Baryon Acoustic Oscillation and Weak Lensing Measurements.* Astrophys. J., 690:923–936 (2009). arXiv:0806.0937.

[97] S. Lee and K.-W. Ng. *Growth index in an accelerating universe.* ArXiv e-prints (2009). arXiv:0905.1522.

[98] H. Wei and S. N. Zhang. *How to distinguish dark energy and modified gravity?* Phys. Rev. D, 78(2):023011–+ (2008). arXiv:0803.3292.

[99] E. Bertschinger and P. Zukin. *Distinguishing modified gravity from dark energy.* Phys. Rev. D, 78(2):024015–+ (2008). arXiv:0801.2431.

[100] P. J. E. Peebles. *Principles of physical cosmology.* Princeton Series in Physics, Princeton, NJ: Princeton University Press (1993).

[101] L. Wang and P. J. Steinhardt. *Cluster Abundance Constraints for Cosmological Models with a Time-varying, Spatially Inhomogeneous Energy Component with Negative Pressure.* Astrophys. J., 508:483–490 (1998). arXiv:astro-ph/9804015.

[102] E. V. Linder and A. Jenkins. *Cosmic structure growth and dark energy.* MNRAS, 346:573–583 (2003). arXiv:astro-ph/0305286.

[103] B. A. Bassett, B. Nichol, and D. J. Eisenstein. *WFMOS: Sounding the dark cosmos.* Astronomy and Geophysics, 46:26–5 (2005). arXiv:astro-ph/0510272.

[104] B. A. Bassett. *Optimizing cosmological surveys in a crowded market.* Phys. Rev. D, 71(8):083517 (2005). arXiv:astro-ph/0407201.

[105] D. Huterer and M. S. Turner. *Probing dark energy: Methods and strategies.* Phys. Rev. D, 64(12):123527 (2001). arXiv:astro-ph/0012510.

[106] A. Rassat, A. Amara, L. Amendola, et al. *Deconstructing Baryon Acoustic Oscillations: A Comparison of Methods.* ArXiv e-prints (2008). arXiv:0810.0003.

[107] B. A. Bassett and R. Hlozek. *Dark Energy*, chapter Baryon Acoustic Oscillations. Cambridge University Press (2009).

[108] M. Kunz. *Why we need to see the dark matter to understand the dark energy.* ArXiv e-prints, 710 (2007). arXiv:0710.5712.

[109] C. Clarkson, M. Cortês, and B. Bassett. *Dynamical dark energy or simply cosmic curvature?* JCAP, 8:11–+ (2007). arXiv:astro-ph/0702670.

[110] R. Hlozek, M. Cortês, C. Clarkson, et al. *Dark energy degeneracies in the background dynamics.* General Relativity and Gravitation, 40:285–300 (2008). arXiv:0801.3847.

[111] E. V. Linder. *Curved space or curved vacuum?* Astroparticle Physics, 24:391–399 (2005). arXiv:astro-ph/0508333.

[112] Z.-Y. Huang, B. Wang, and R.-K. Su. *Uncertainty on Determining the Dark Energy Equation of State due to the Spatial Curvature.* International Journal of Modern Physics A, 22:1819–1834 (2007). arXiv:astro-ph/0605392.

[113] *http://www.cosmology.org.za.*

[114] *http://www.mathworks.com/matlabcentral/fileexchange/.*

[115] A. Cooray, D. Huterer, and D. Baumann. *Growth rate of large-scale structure as a powerful probe of dark energy.* Phys. Rev. D, 69(2):027301–+ (2004). arXiv:astro-ph/0304268.

[116] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. *The complex-step derivative approximation.* ACM Trans. Math. Softw., 29(3):245–262 (2003).

[117] H.-J. Seo and D. J. Eisenstein. *Improved Forecasts for the Baryon Acoustic Oscillations and Cosmological Distance Scale.* Astrophys. J., 665:14–24 (2007). arXiv:astro-ph/0701079.

[118] Y. Wang. *Dark Energy Constraints from Baryon Acoustic Oscillations.* Astrophys. J., 647:1–7 (2006). arXiv:astro-ph/0601163.

[119] *http://cmb.as.arizona.edu/~eisenste/acousticpeak/bao_forecast.html.*

[120] *http://www.mathworks.com/matlabcentral/fileexchange/.*

[121] E. V. Linder. *Exploring the Expansion History of the Universe.* Phys. Rev. Lett., 90(9):091301–+ (2003). arXiv:astro-ph/0208512.

[122] *http://www.mpa-garching.mpg.de/galform/millennium/.*

[123] *http://map.gsfc.nasa.gov/.*

[124] Read the BSD license available in the Fisher4Cast distribution, or go to http://www.opensource.org/licenses/bsd-license.php for more information.